CELL FORMATION - ALGORITHMS
AND COMPARISONS

A Thesis Submitted

in Partial Fulfilment of the Requirements

for the Degree of

MASTER OF TECHNOLOGY
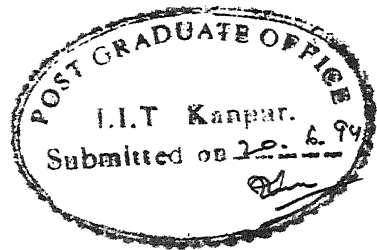
by

JAYESH SHARMA

to the

INDUSTRIAL AND MANAGEMENT ENGINEERING PROGRAMME

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
JUNE, 1994

CERTIFICATE

It is certified that the work contained in the thesis entitled " Cell Formation - Algorithms and Comparisons ", by Jayesh Sharma, has been carried out under my supervision, and that this work has not been submitted elsewhere for a degree.

(Dr. Kripa Shanker)
Professor
Industrial & Management Engineering
Indian Institute of Technology
Kanpur 208 016

June, 1994

ABSTRACT

The rapid growth on technological frontiers and increasing competition in the market has adversely influenced the manufacturing plants turning out production in mass or batches of large volume. More and more companies are submitting to group technology and flexible manufacturing systems. The benifits accrued from a change over to group layout has made cellular manufacturing attractive to both academicians and practitioners. This study addresses the problem of cell formation in cellular manufacturing.

Numerous heuristic and analytical methods have been developed and applied in literature, but very little effort has been put into comparing and selecting the best or appropriate method. Solution of practical problems without adequate generalizations and theoretical formulations is limitedly useful. When researchers and practitioners claim success with a certain method it is necessary to compare and evaluate result on absolute quantitative scale.

The present work compares some selected algorithms on the basis of grouping efficiency, grouping efficiacy, measure of effectiveness, time, number of bottleneck machines and number of exceptional parts.

ACKNOWLEDGEMENT

CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

A	Machine Component Incidence Matrix
a_{ij}	(i,j)th Element of the Machine & Part Incidence Matrix
$A^{(k)}$	Matrix A at Iteration k
BEA	Bond Energy Analysis
BM	Bottleneck Machines
C_r	Machine Group
CAM	Computer Aided Manufacturing
CFA	Component Flow Analysis
CIA	Cluster Identification Algorithm
CIMS	Computer Integrated Manufacturing systems
CM	Cellular Manufacturing
D_r	Diagonal Submatrices
DCA	Direct Clustering Algorithm
e	Total no. of 1s in Matrix
e_0	No. of Nonzero Elements Outside the Diagonal Block
e_d	No. of Non Zero Elements in the Diagonal Blocks
e_v	No. of Voids in Diagonal Cells
EP	Exceptional Parts
FMS	Flexible Manufacturing System
GE	Grouping Efficiency
GEF	Grouping Efficiacy
GT	Group Technology
$I_1 \; I_2$	Set of Machines
IME	Improvement in Measure of Effectiveness
J	Set of Components

JIT	Just in Time
LWA	Linear Weighting Algorithm
m	No. of Machines
M_j	No. of Machines in Column j
M_r	No. of Machines in rth Cell
MC-k	Machine Cell k
MRP	Material Requirements Planning
n	No. of Components
N_r	No. of Parts in rth Cell
OV	Occupancy Value
OVM	Occupancy Value Method
P_1	Route of Component 1
P_r	Product Group r
PF-k	Part Family k
PFA	Production Flow Analysis
q	Weight
ROC	Rank Order Clustering
ROC2	Extended Rank order Clustering
η, η_1, η_2	Efficiency

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Introduction

Over the last two decades, group technology (GT) has emerged as an important scientific principle (Kusiak [41]) in improving the productivity of manufacturing systems. Cellular manufacturing (CM) / GT can be called as one of the most important innovations of the last few years. More and more manufacturing industries involved with small lots and a variety of products are becoming interested in group technology, which is particularly applicable in the area of batch type manufacturing.

Group technology has also been recognized as an essential element of the foundation for the successful development and implementation of computer aided manufacturing (CAM) through the application of the part family concept. Recognizing the contribution that GT can make to the implementation of Computer Integrated Manufacture (CIM), Merchant says, " the appropriate initial step, to lay a sound foundation for the gradual evolution of a factory to full computer control, is to institute group technology cellular organization". The application of group technology does not depend upon the degree of automation, and may be applied in a totally automated or even in a manual production system (Choobineh [19]). The basis for implementation of cellular manufacturing systems is the allocation of parts or components with certain 'similarities' into groups or families, and the

formation of machining cells which are capable of manufacturing one or more part families. Manufacturing control can then be exercised within or between cells, rather than at the individual machines, using a combination of computer controlled machining and handling devices.

Group technology has probably had a greater impact on increasing manufacturing productivity' than any other manufacturing concept. This can be attributed partly to contributions made by cellular manufacturing concepts to other manufacturing technologies such as roboticised and flexible manufacturing systems.

Group technology is a manufacturing philosophy in which similar parts are identified and grouped together to take the advantages of their similarities in manufacturing and design. The idea behind GT is to decompose a manufacturing system into subsystems. In addition to the simplification of management control through the creation of similar sub systems called cell, GT also leads to many other benifits. A cell is essentially a group of machines and a family of related components. For example, for a plant producing 10,000 different parts it may be able to group the vast majority of these parts into 50 or 60 distinct families having similar design and/or manufacturing characteristics. This leads to similar processing of members and high manufacturing efficiencies. The cell is self contained and self managed, with performance control entrusted to the foreman who wields authority to change loading on machines, assigns

operators to machines or alter the sequence of operations that he deem fit for proper load balancing within cell so long as such decisions do not contradict the overall strategies laid down by top management.

The part families are assigned to cells to satisfy some of the following objectives

(1) Minimize total material handling cost which includes costs due to inter cellular moves and intra cellular moves.

(2) Maximize within cell utilization of machines.

(3) Minimize duplication of machines in different cells.

(4) Minimize number of exceptional parts and inter cellular trips.

With more effective design rationalization and manufacturing standardization, the efficiency associated with flow line production is sought while the flexibility of a job shop manufacturing system is maintained. These features are essential for a firm to remain competitive in the current manufacturing environment, in which more special orders are demanded, production life cycles are reduced, and competition is focussed on factors like delivery speed, quality, design flexibility, delivery reliability as well as price. Firms have to manufacture products in a larger product mix, smaller volume, increased part complexity and shorter production period. Group technology, with its promises to reduce lead time, material handling, set up time, expediting, work in process and finished goods inventories, was developed as an approach to address the difficulties faced by

today's manufacturing industry. More over the need for flexibility and just in time processing capabilities is forcing many discrete parts manufacturers to consider reorganizing their facilities.

The vast majority of modern engineering factories are required to manufacture many different products which are ordered irregularly and usually small quantities. The product range is usually wide and the company is forced to manufacture many thousands of different components. In such companies heaps of components are seen lying around the workshop floors. Components which are urgently required hold up the assembly stages of production.

The inefficiency and disadvantages encountered for large variety and medium/small batch size production in the traditional manufacturing environment pose three important questions

(1) What should be the type and number of facilities ?

(2) What should be the layout of facilities ?

(3) How should parts be scheduled and released for production ?

The answers to the above three questions lie in the emerging concepts of group technology/cellular manufacturing system (CMS) and flexible manufacturing system.

1.2 Implementation of group technology

A gradual implementation of GT in stages in a factory is more commonly adopted when compared to in one step, because of
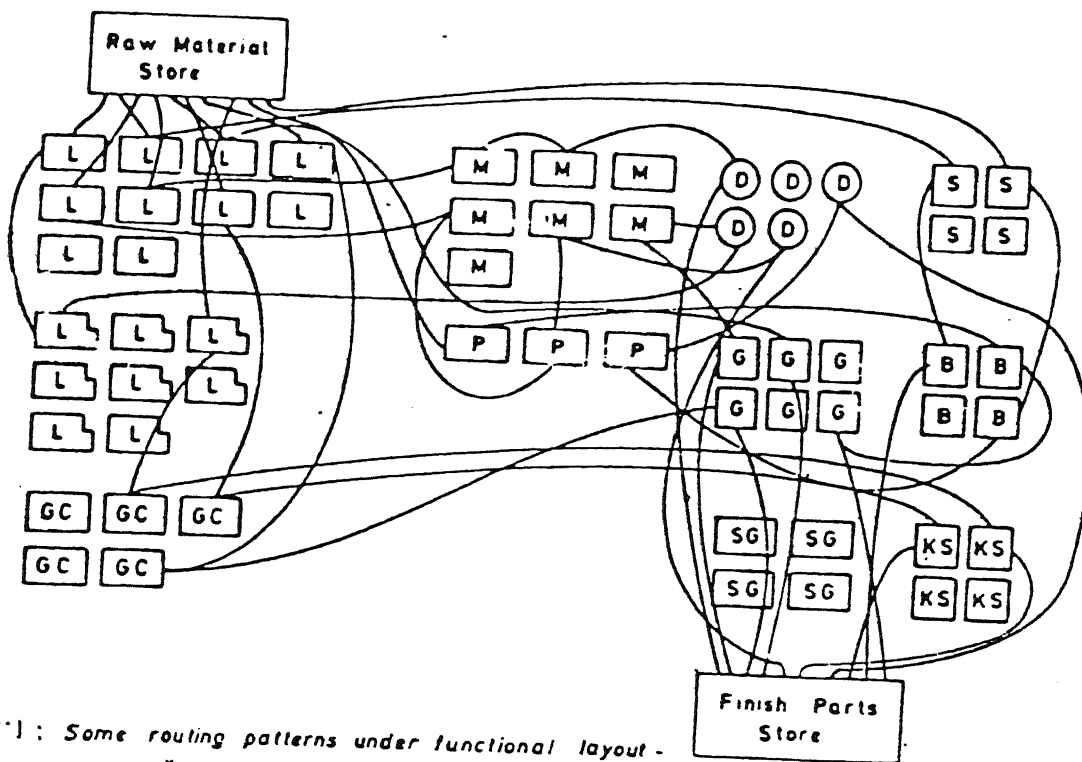
Fig 1·1 : Some routing patterns under functional layout -
"A bowl of spaghetti "

Figure 1.2 - Group Layout
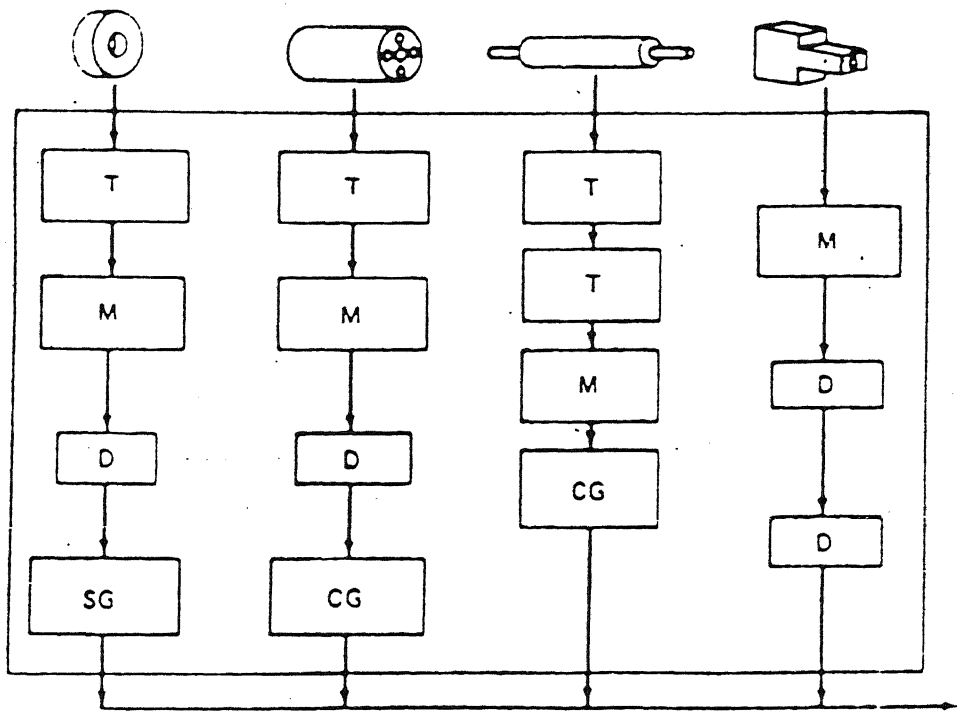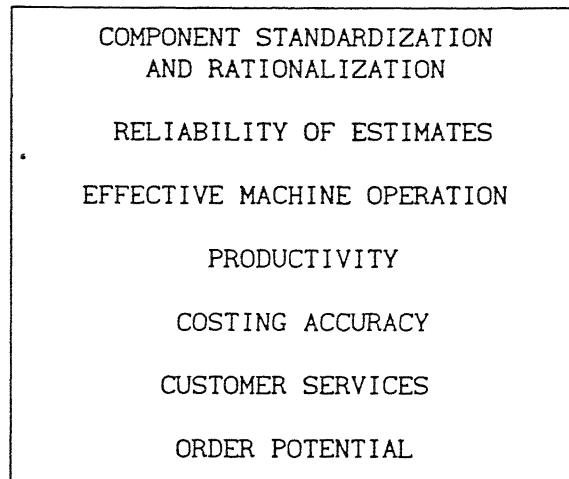
```
┌─────────────────────────────────────┐
│   COMPONENT STANDARDIZATION          │
│      AND RATIONALIZATION             │
│                                      │
│    RELIABILITY OF ESTIMATES          │
│                                      │
│  EFFECTIVE MACHINE OPERATION         │
│                                      │
│         PRODUCTIVITY                 │
│                                      │
│        COSTING ACCURACY              │
│                                      │
│       CUSTOMER SERVICES              │
│                                      │
│        ORDER POTENTIAL               │
└─────────────────────────────────────┘

          ┌──────────────┐
          │ CAN INCREASE │
          └──────────────┘

        ┌────────────────────┐
        │  GROUP TECHNOLOGY  │
        └────────────────────┘

          ┌──────────────┐
          │  CAN REDUCE  │
          └──────────────┘

┌─────────────────────────────────────┐
│         OVERALL COST                 │
│                                      │
│      FINISHED PARTS STOCK            │
│                                      │
│    OVERALL PRODUCTION TIMES          │
│                                      │
│         WORK MOVEMENT                │
│                                      │
│        WORK IN PROGRESS              │
│                                      │
│           DOWN TIME                  │
│                                      │
│          SETTING TIME                │
│                                      │
│           PAPER WORK                 │
│                                      │
│         PLANNING EFFORT              │
└─────────────────────────────────────┘
```
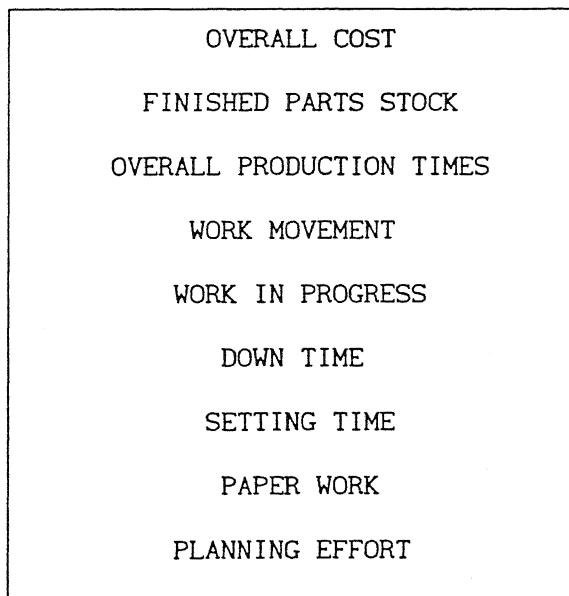
Figure 1.3 : Advantages of Group Technology

the inherent risks and disruptive effects to total re-organization. A GT programme could require 2-3 years to install. Generally, GT implementation team is created whose members are drawn from a variety of departments such as production, production planning and control and design. First a pilot line is established until a satisfactory long term solution is reached which suits the unique circumstances of the factory. At the same time existance of cell enables all members of the factory to become slowly accustomed to the new manner of work and re-organize the advantages. Additional cells are slowly introduced one by one till all components are manufactured within the cells.

1.3 Scope of present work

During the past decade, there has been tremendous interest in GT problem from practitioners as well as from academics. Numerous heuristic or analytical methods have been developed and applied, but very little effort has been put into selecting and comparing the best or appropriate method.

Solution of practical problems without adequate generalizations and theoretical formulations is limitedly useful. When researchers and practitioners claim success with a certain method it is necessary to compare and evaluate result on absolute quantitative scale.

Present work codes various algorithms and compares them on similar quantitative and qualitative scales.

1.4 Organization of the thesis

Chapter I gives the introduction to the subject of group technology. The advantages and disadvantages of group technology are mentioned. The explanation of 'How to implement group technology' is given.

Chapter II gives the detailed literature survey. A brief observation on literature survey is made towards the end of the chapter.

Chapter III explains various algorithms which are coded and compared in this work.

Chapter IV describes various measures of comparisons on the basis of which various algorithms are compared. This chapter explains the concept of Grouping efficiancy, grouping efficiacy, measure of effectiveness, bottleneck machines and exceptional parts.

Chapter V lists the observations and comparisons done while solving the illustrations with various methods. The explanation for the occurance of the observations is also given. Four illustrations, relevent tables and graphs are also given in this chapter.

Chapter VI gives the conclusions and the suggestions for the further work.

Appendix A gives the calculations of the various methods explained in the chapter 3. It also shows the calculations for calculating grouping efficiancy and grouping efficiacy.

CHAPTER II

LITERATURE SURVEY

In this chapter a brief description of some of the major efforts in the field of group technology is presented. Since group technology aims at dividing the production systems, most of the research is focused on dividing the machine and components into groups. To understand what we mean by dividing of production system, following example is given.

Consider the machine part incidence matrix (Fig 2.1) in which $a_{ij} = 1$ indicates that component j visits machine i, and a_{ij} is blank, otherwise.

0	1	2	3	4	5
1		1		1	1
2	1		1		
3		1			
4	1		1		

Figure 2.1 : Machine Part Incidence Matrix

By rearrangement of rows and columns, if some method can transform Fig 2.1 into matrix Fig 2.2 then we say that production system is divided into groups.

0	1	3	2	4	5
2	1	1			
4	1	1			
1			1	1	1
3			1	1	

Figure 2.2 : Block Diagonal Structure

The organization of description of major efforts in the field of group technology, is based on classification scheme. The efforts can be classified under approaches like

(1) Rule of thumb

(2) Classification and coding system

(3) Flow analysis

(4) Approaches using similarity coefficient

(5) Cluster analysis

(6) Mathematical programming approach

(7) Heuristic approaches

(8) Hybrid approaches

(9) Fuzzy theoretic approaches

(10) Expert System / Artificial Intelligence approach

2.1 Rule of thumb

Based on the production engineer's local product knowledge of the parts and processes, some families of the components may be self evident. Families consist of components defined by name or functions e.g pump casings, shafts, and sleeves.

2.2 Classification and coding system

A classification and coding scheme sorts parts into different classes based on certain part characteristics (such as routings) and assigns a code (usually alphanumeric) such that parts having similar codes could be easily identified as parts having similar codes could be easily identified as part families. A number of commercial (for e.g BRISCH, CODE, MICLASS) as well as non-proprietary (for e.g OPITZ, KC - 1) coding schemes are

currently available. Most schemes are useful for variety reduction, and design information retrieval, and rarely have the capability as stand alone procedures for finding part families for cellular manufacturing. This is due to the fact that they group parts which are similar in shape but differ in batch size or tolerances and, hence should be made on different types of machines. Additionally, they do not group parts which are similar in design features but are processed on the same set of machines (Burbidge [9]). There are a few exceptions. The MICLASS system developed by TNO in Holland considers both design as well as part routing features and can be used to form part families and machine cells for cellular manufacturing systems.

2.3 Flow analysis

The aim of flow analysis is that of 'finding the families of components and associated groups of machines for group layout, by a progressive analysis of the information contained in route card'.

2.3.1 Production flow analysis

Production flow analysis (PFA) was first introduced by Burbidge [8]. It involves the systematic listing of the components in various ways in the expectation that groups of machines and components may be found by careful inspection. As Beer and de Witte [22] point out, the procedure requires a series of evaluations to be made by the designer, more or less calling upon his ability to recognize pattern.

PFA consists of three levels of analysis.

(a) Factory flow analysis

It makes use of process route numbers, in order to get an overall picture of the present state of material flows. Machines are divided into departments, and each department is given a number. The process route number of a component is defined as the sequence of the numbers of the departments visited. A flow chart showing the interaction of various departments based on these numbers is then drawn.

(b) Group analysis

It attempts at effective formulation of groups of machines and their families of related components. The 'best' method called 'Nuclear Synthesis' is based on selecting machines used by a few components as starting points for various cells. or nuclei. The next machine is allocated on the basis that it has the smallest number of components left unassigned to a group once 'Nuclear Synthesis' is completed, these nuclei are modified and combined, until the required number of groups is formed.

(c) Line analysis

It aims at creating a pure line-layout between machines inside the groups.

2.3.2 Component flow analysis

Component flow analysis (CFA) was first used by El-Essawy [26]. CFA employs three stages of analysis.

FIRST STAGE : The objective is to consider the total component mix of the company and to identify and sort components into categories according to their manufacturing requirements. The components are sorted in order of machine requirements and sorted list are printed in two ways, firstly in the order of the number of machines required and secondly in the order of the smallest machine numbers involved.

SECOND STAGE : The aim is to obtain groupings of the machines using the lists of sorted components and taking into account various local constraints. Rough groups are formed, to which other machines and components are successively added.

THIRD STAGE : It involves a detailed analysis of the loadings and flow pattern of the cells with appropriate adjustments to ensure that an acceptable design is achieved.

PFA differs from CFA in following ways. PFA first partitions the problem whereas CFA does not. The manner in which the cells were built up is also different in two methods.

2.4 Approaches using Similarity coefficients

The basis of this method is to measure the 'similarity' between each pair of machines and then to group the machines into families based on their similarity measurements. In most cases, the similarity measurement used is the coefficient of Jaccard which is defined for any pair of machines as 'the number of components which visit both machines, divided by the number of components which visit at least one of the machines'. The components having similar coefficients are grouped together to

form a cell.

Of the several similarity coefficients selection of a particular similarity coefficient depends upon its application.

M   : Total no. of machines in the system.

A   : Number of machines used by both components.

B   : Number of machines used by one of the component.

C   : Number of machines used by the other component.

D   : Number of machines not used by either of the components.

ASC_{ij} : Additive type of similarity coefficient between i th and j th machines.

SC   : Similarity Coefficient.

e_{ij} : Number of common components using both machines i and j.

TNC_i : Total number of components using i th machine.

SC'_{ij} : Similarity coefficient of product type based on total number of common components processed by each of the machines i and j.

E_i : Total number of common components processed by i th machine.

Similarity coefficient can be defined as

SC = A / (A + B + C)   (Waghodekar and Sahu [68])   (2.1)

$ASC_{ij} = e_{ij}$ / $(TNC_i + TNC_j - e_{ij})$ (2.2)

(Rajagopalan and Batra [54])

Other similarity coefficients can be defined as

(a) SC = A / M $\qquad$ (2.3)

(b) SC = A / (B + C) $\qquad$ (2.4)

(c) SC = 2A / (2A + B + C) $\qquad$ (2.5)

(d) SC = B + C / M $\qquad$ (2.6)

(e) SC = (A + D) / (B + C) $\qquad$ (2.7)

Other similarity coefficients can be defined as

(a) $SC'_{ij} = (e_{ij})^2 / (TNC_i * TNC_j)$ $\qquad$ (2.8)

$\qquad$ (if E_i, $E_j = 0$, assume $SC_{ij} = 0$).

(b) $SC_{ij} = (e_{ij})^2 / (E_i * E_j)$ $\qquad$ (2.9)

McAuley [47] uses single linkage cluster analysis. This method first clusters together those machines mutually related with the highest possible similarity coefficient, then it successively lowers the level of admission by steps of predetermined equal magnitude. The admission of a machine or groups of machines into another group is by a criteria of single linkage. The main disadvantage of this method is that while two clusters may be linked by this technique on the basis of a single bond, many of the members of the two clusters may be quite far removed from each other in terms of similarity.

Rajagopalan and Batra [54] developed a graph theoretic method which uses cliques of the machine-graph as a means of classification. The vertices of this graph are the machines and the arcs are the Jaccard similarity coefficients and a clique is a maximal collection of vertices, every pair of which is connected by an edge of the graph. The main disadvantage of this

approach is that because of the high density of the graph, a very large number of cliques is usually involved and many of the cliques are not vertex disjointed. To reduce the number of groups and to incorporate the machines which are not included in the cliques, graph partitioning is used and it is at this stage that the allocation of components, in accordance with a number of heuristic rules, is also carried out.

2.5 Cluster analysis

Cluster analysis is the widely used technique for cell formation in group technology. These methods re-arrange the rows and columns of the machine component incidence matrix such that the 1's are brought together. Each block of 1's constitutes a component family and a machine cell.

King [39] proposed a Rank order clustering (ROC) technique based on cluster analysis. This method interprets each entry in the matrix as a binary word and computes decimal equivalent of each row and column. First rows and then columns are sorted in decreasing order of the binary word values, until algorithm converges.

Chandrasekharan and Rajagopalan [15] observed that ROC does not provide block diagonal structure even in case of structured matrices where such a possibility exists. They found that ROC is dependent on the initial disposition of the incidence matrix and that it does not handle exceptional elements too well. They suggested MODROC.

King and Nakornchai [40] observed that reading entries as binary words restricted the size of the problem to less than 47 machines and 47 components because the largest integer representation in many computers was $(2^{48} - 1)$. They proposed ROC2 which is computationally efficient.

Chan and Milner [13] proposed direct clustering algorithm which is a poor version of ROC algorithm, except that it eliminates the sensitivity of the latter to the configuration of the initial matrix. This algorithm has same handicaps as the ROC algorithm and uses a limited binary comparison procedure for ranking.

Graham [29] proposed linear weighting algorithm where the weights are increased linearly. The i th row is given a weighting of $m - i + 1$ where m is the total number of rows, and the priority ranking value is determined as the mean of the weightings of the non-zero entries. Ranking values calculated this way can be found and sorted very quickly and the requirement of a very large integer representation does not arise. The major disadvantages of this linear weighting algorithm are the complicated and very confusing patterns of the intermediate results together with the difficulty in predicting the behavior of the procedure.

McCormick [48] proposed Bond energy algorithm which reorders the rows and columns of the matrix for the purpose of moving

numerically larger matrix elements together. A measure of closeness between two component vectors or machine vectors termed bond energy is defined. Method seeks to determine permutation of rows and columns in which the sum of products of adjacent elements is maximized. This is a restricted form of the quadratic assignment problem.

Khator and Irani [38] proposed a heuristic procedure, the occupancy value method, for identifying clusters in a machine component matrix created from route card data. A unique feature of this method is that it progressively develops block diagonalization starting from the northwest corner of the matrix.

Boe and Cheng [7] identified several deficiencies of clustering and array based sorting algorithms and suggested a clustering algorithm 'close neighbour algorithm'. In stage one the clustering of machines is performed and in stage two clustering of parts is done. The algorithm converges in one pass in most of the problems.

Kusiak [41] proposed cluster identification algorithm (CI) which identifies the cluster if it exists. He proposed three algorithms based on CI to solve GT problem with bottleneck components and machines. The first algorithm solves an unconstrained GT problem. The second heuristic considers a constraint restricting the number of machines in each cell and identifies the bottlenecks.

Tam [64] proposed a new similarity coefficient based on the operation sequence and used a clustering technique to obtain the component families. Such a coefficient, augmented with an advanced clustering technique, has been shown to improve productive effectiveness.

2.6 Mathematical programming approaches

Using mathematical approach of operations research (Linear programming, zero-one integer programming and dynamic programming), the GT cell production problem may be given concise mathematical formulation so that a solution can be found for a given constraint set.

Gunasingh and Lashkari [30,31] proposed two zero-one integer programming formulations for the GT problem. The first formulation groups machines based on the compatibility of processing components, while the second formulation groups machines in order to minimize the cost of machine allocation and the cost of intercellular movement.

Gunasingh and Lashkari [32] presented a non-linear 0-1 integer programming formulation to simultaneously group machines and parts in cellular manufacturing systems based on tooling requirements of the parts, tools available on the machines and the processing times. The formulations take into account the limitations on the number of parts and machines in a group and the number of machine types available.

Boctor [6] presented 0-1 integer programming based formulation for cell formation with the objective of minimizing the exceptional elements. This model allows the designer to control the cell sizes.

Kusiak [43] developed a integer programming problem. He considered presence of alternative process plans and suggested a 0-1 integer program based p-median model to determine the optimal process plan and assignments of parts to machine cells. Incorporating process plans will result in an improved quality of part families and the machine cells.

Wei and Gaither [71] suggested an alternative formulation based on 0 - 1 integer programming for cell formation problem with the objective of minimizing the cost of manufacturing exceptional parts subject to constraint on the number of cells, cell size and the machine capacities.

Kumar [66] have formulated a joint grouping and loading problem as a multistage multi objective model.

Shtub [58] showed that the generalized cell formation problem due to Kusiak [41] is equivalent to the generalized assignment problem.

Choobineh [19] developed an integer programming problem to determine the assignments of component families to machine cells with the objective of minimizing the sum of the total production

cost and cost of purchasing new equipment. The model assumes that more than one component family be assigned to a cell.

Srinivasan [60] proposed an assignment model for GT. The distance matrix for machines is solved as an assignment problem from which initial cells are identified. Component families are similarly identified and they are merged using a set of rules.

2.7 Heuristic approaches

Heuristic approaches have the ability to incorporate several practical considerations simultaneously.

Waghodekar and Sahu [68] proposed an algorithm called machine component cell formation (MACE) to solve the grouping problem. The main feature of this heuristic is that it aims at minimizing the number of exceptional elements.

Ballakur and Steudel [4] developed model which considers objectives such as minimizing exceptional elements, job-lateness and maximizing machine utilization imposing an upper limit on the number of machines in the cell.

Askin and Subramaniam [2] developed a clustering algorithm that considers manufacturing costs like fixed and variable machining cost, setup cost, production cycle inventory cost, work in process inventory cost, material handling cost. The algorithm consists of three stages. In stage 1 parts are classified using a coding system. In stage 2 an attempt is made to develop a feasible grouping of parts based on the manufacturing costs. In

stage 3 the actual layout of machine cells is analyzed.

Offodile [50] suggested a cell formation approach which aims at identifying parts with similar processing requirements.

Harhalakis [35] described a twofold heuristic algorithm aimed at minimizing inter cell material movement. The first stage of the proposed heuristic is a bottom-up aggregation procedure to minimize the normalized inter cell traffic defined as the ratio of the inter cell movement between any two cells under consideration and the number of machines included in these cells. The second stage is aimed at improving the solution obtained from the first stage.

Logendran [44] proposed a model which incorporates both intra cell and inter cell moves. Logendran [45] considered sequence of operations into account in the design of cellular manufacturing system. This model includes the impact of cells in evaluating inter cell moves. The model also aims at maintaining a targeted utilization of work station.

Gupta and Seifoddini [33] proposed a methodology which incorporates production volume of various part types, routing sequence and unit operation times through an appropriately defined similarity coefficient. The first stage determines machine groups. The second stage assigns each part to one of the cells.

Sule [63] outlined the importance of machine capacity planning in GT environment. Sule's method determines the number of machines, their groupings and the amount of material transfer so that the cost of processing the components within the plant is minimized.

2.8 Hybrid approaches

The literature survey till now has shown that certain cell formation methods employs both heuristic and mathematical procedures in different stages. Such methods are called hybrid approaches.

Steudel and Ballakur [4] proposed a two stage heuristic in which the first stage employs a dynamic programming procedure to generate an optimum chain of machines in which the sum of bonds between machines in the chain is maximized. The second stage is a heuristic, that partitions the chain to form machine cells subject to cell size restrictions.

Nagi [49] addresses two problems (a) Process plan selection (formulated as a linear programming problem). (b) Cell formation, is solved through heuristic. The method also considers the sequence of operations and production data such as production volume, processing time, machine capacity.

Choobineh [19] employed a two stage procedure. First employs a similarity measure based on part operations and operation sequences and attempts to identify part families using a

heuristic procedure. The second stage formulates an integer programming which specifies the type and number of machines in each cell and the assignment of part families to the cells.

Askin and Chiu [1] proposed a mathematical model which addresses the problem of grouping of individual machines into cells and the routing of components to machines within cells. The mathematical programming formulation incorporates costs of inventory, machine depreciation, machine set-up and material handling costs. The formulation is then divided into two subproblems. The heuristic graph partitioning procedure has been proposed for each subproblem. The first subproblem assigns components to specific machines. The second sub problem groups machines into cells. They have also presented an approach for determining economic batch sizes in the group technology environment.

2.9 Fuzzy theoretic approaches

It was first advocated by Rajagopalan [54]. Chu and Hayya [20] observed that fuzzy approach reveals part families and also associates a degree of closeness between part and its part family. This information helps the designer in determining to which part family should be assigned as to balance workload among machine cells.

2.10 Observations

Following points were observed while doing literature survey.

(1) Very less work has been done to integrate GT with materials requirements planning, just in time, computer integrated

manufacturing systems and with assembly operations. In my view such integration is essential to realize the benifits of group technology.

(2) Very less efforts have been made to incorporate flexibility and flexibility measures in the design of group technology.

(3) Group technology involves multiple objectives. Less efforts has been done to use mathematical techniques like goal programming in group technology.

(4) Less work has been done in developing probabilistic approaches in the design of group technology.

(5) There are many approaches available for design of group technology. The problem now arises is of finding which method is suitable for what type of problem. There is a need for comparison of group technology designs with this problem in mind.

(6) Less work has been done to economically justify the methods of group technology.

(7) Less work has been done to find out managerial implications on group technology.

(8) Recent research is tending towards the application of artificial intelligence techniques and expert systems.

CHAPTER III

SOME ALGORITHMS ON CELL FORMATION

In this chapter some of the algorithms which are selected for subsequent comparison purpose are explained. These algorithms are both array based and non-array based heuristic algorithms. The same algorithms are coded for the purpose of comparison. The array based algorithms are

(a) Rank order clustering (King [39])

(b) Linear weighting algorithm (Graham [29])

(c) Direct clustering algorithm (Chan and Milner [13])

(d) Extended rank order clustering (King and Nakornchai [40])

(e) Bond energy algorithm (McCormick et. al. [48])

and the heuristic algorithms are

(a) Cluster identification algorithm (Kusiak [41])

(b) Occupancy value method (Khator and Irani [38])

3.1 Rank order clustering (ROC)

ROC is a well known clustering technique that attempts to create a block diagonal form by repeatedly reallocating the columns and rows of a machine part matrix according to binary values. King [39] shows that if the patterns of row entries are read as binary words then they can be ranked in reducing binary value order. using this principle ROC represents route card data as a binary matrix. Using a positional weighting technique for the '1' entries in the matrix, the rows and columns are alternately rearranged in order of decreasing rank. The result is

a diagonalization of the 1's into several clusters. If independent machine component groups do exist in the sample data provided, each machine will occur in only one cluster. Components will be uniquely assigned to any one of the clusters. Using this algorithm, the analyst can obtain a visual assessment of the machine groups and the associated families of parts simultaneously. With such an approach, a very valuable preliminary assessment of machines can be obtained because if a large number of machines are shared over several clusters, plans for cellular manufacture can be shelved at the out set.

ALGORITHM

STEP 1 : For each row of the machine-part incidence matrix, assign binary weight and calculate a decimal equivalent.

$$\text{For row i} \quad : \quad \sum_{k=1}^{n} a_{ik} * 2^{n-k} \qquad (3.1)$$

$$\text{For column j} \quad : \quad \sum_{k=1}^{m} a_{kj} * 2^{m-k} \qquad (3.2)$$

m = no. of machines and n = no. of parts.

STEP 2 : Sort rows of the binary matrix in decreasing order of the corresponding decimal weights.

STEP 3 : Repeat the preceding two steps for each column.

STEP 4 : Repeat the preceding steps until the position of each

element in each row and column does not change.

The steps of the algorithm are explained in Appendix through an illustration (Appendix A.1).

3.2 Linear weighting algorithm

Another approach for clustering data was proposed by Graham [29]. In this method the weights increases linearly according to the positions of the rows or columns. The i'th row is given a weighting of m - i + 1 whereas j'th column is given a weighting of n - j + 1. The priority ranking value is determined as the mean of the weightings of the non zero entries. Ranking values calculated this way can be found and sorted very quickly and the requirement of a very large integer representation does not arise.

ALGORITHM

STEP 1 : For each row of machine part incidence matrix assign linear weight. Calculate the weighted sum.

$$\text{For row } i \; : \; \sum_{k=1}^{n} a_{ik} \; * \; (n - k + 1) \qquad (3.3)$$

$$\text{For column } j \; : \; \sum_{k=1}^{m} a_{kj} \; * \; (m - k + 1) \qquad (3.4)$$

STEP 2 : Sort rows in decreasing order of corresponding weights.

STEP 3 : Repeat preceding steps until position of each element in each row and column does not change.

The steps of algorithm are explained in Appendix through an illustration (Appendix A.2).

3.3 Direct clustering algorithm

The Direct clustering algorithm as proposed by chan and milner [13] is a poor version of the ROC algorithm, except that it eliminates the sensitivity of the latter to the configuration of the initial matrix. The number of positive cell entries K in each row and column is counted. The input machine component matrix is rearranged with columns in decreasing order of K but rows in increasing order of K. So, appropriate diagonalization is created and the machine component matrix is input in the same format always to the algorithm.

ALGORITHM

STEP 1 : Determine total number of 1s in each row and column in machine part incidence matrix.

$$\text{For row i} \quad : \sum_{k=1}^{n} a_{ik} \quad\quad (3.5)$$

STEP 2 : Sort each row in decreasing order corresponding to the total number of ones.

STEP 3 : Sort each column in decreasing order corresponding to

the total number of ones.

STEP 4 : Repeat the preceding steps until the position of each element and column does not change.

The steps of algorithm are explained in Appendix through an illustration (Appendix A.3)

3.4 Bond energy analysis (BEA)

McCormick et. al. [48] developed a matrix clustering technique which they call the bond energy algorithm. The matrix is applicable to any matrix in which non-negative integer values of an element in the matrix express a measure of the degree of association of the corresponding rows and column entities. The bond energy analysis algorithm attempts to identify and exhibit the interrelations within each cell and the associations among the clustered groups by means of total bond energy. A bond is claimed to exist between each pair of the neighboring rows and columns if they have positive cells in the machine-part matrix. BEA begins with an arbitrarily selected column (or rows). It then places that column with the greatest contribution to the total bond energy beside the assigned column (row). It repeats the same procedure for all the columns and rows. The BEA seeks to form a block diagonal form by maximizing the measure of effectiveness, which is defined as follows

$$ME = 1 / 2\{ \sum_1^m \sum_1^n a_{ij} * (a_{i,j-1} + a_{i,j+1} + a_{i,j+1}$$
$$a_{i-1,j} + a_{i+1,j})\} \qquad (3.6)$$

ALGORITHM

STEP 1 : Set j = 1. Select one of the columns arbitrarily.

STEP 2 : Place each of the remaining n - j columns, one at a time, in each of j + 1 positions, and compute each columns contribution to the ME. Place the column that gives the largest incremental contribution to the ME in the best location. Increase j by 1 and repeat the preceding steps until j = n.

STEP 3 : When all the columns have been placed, repeat the procedure for the rows.

The steps of algorithm are explained in Appendix through an illustration (Appendix A.4).

3.5 Extended Rank order clustering algorithm

King and Nakornchai [40] observed that procedure of reading the entries as binary words presents some computational difficulties. Since the largest integer representation in most computers is $(2^{48} - 1)$ or less, the maximum number of rows or columns that could be dealt with this way would be 47. They further observed that incidence matrices of the kind involved in group technology problems are usually very sparse, with densities unlikely to be higher than 5 to 10 %. This means that an elaborate system of linked list structures would in general be economical.

King and Nakornchai [40] proposed new extended ROC algorithm

where the whole sorting procedure is reduced to that of shifting the orders of rows and columns.

ALGORITHM

STEP 1 : For each column, locate the rows (machines) with entries, move the rows with entries to the head of the row list, maintaining the previous order of the entries.

STEP 2 : For each row, locate the columns with entries, move the columns with entries to the head of the column list, maintaining the previous order of the entries.

STEP 3 : Repeat previous steps until there is no change in the matrix.

The steps of algorithm are explained in Appendix through an illustration (Appendix A.5)

2.6 Cluster identification algorithm

In 1968 Iri [37] suggested masking technique. Starting from any row, mask all the columns which have an entry in this row, then proceed to make all rows which have entries in these columns. Repeat the process until the numbers of rows and columns stop increasing. These rows and columns constitute a block.

In 1987 Kusiak and Chow [41] applied the concept presented in Iri [37] to develop the cluster identification algorithm (CIA). The cluster identification algorithm allows one to check

the existence of mutually separable clusters in a binary machine part incidence matrix provided they exist.

ALGORITHM

STEP 0 : Select iteration number k = 1.

STEP 1 : Select any row i of incidence matrix $A^{(k)}$ and draw horizontal line h_i through it.

STEP 2 : For each entry of 1 crossed by the horizontal line h_i draw a vertical line v_j.

STEP 3 : For each entry of 1 crossed once by a vertical line v_j draw a horizontal line h_k.

STEP 4 : Repeat steps 2 and 3 until there are no more crossed once entries of 1 in $A^{(k)}$. All crossed twice entries of 1 in $A^{(k)}$ form machine cell MC-k and part family PF-k.

STEP 5 : Transform the incidence matrix $A^{(k)}$ into $A^{(k+1)}$ by removing rows and columns corresponding to all the horizontal and vertical lines drawn in steps 1 through 4.

STEP 6 : If matrix $A^{(k+1)} = 0$, where 0 denotes a matrix with all elements equal to zero, stop, otherwise set k = k + 1 and go to step 1.

The steps of the algorithm are explained in Appendix through

an illustration (Appendix A.6).

3.7 Occupancy value method

Khator and Irani [38] proposed an heuristic approach to machine component grouping in cellular manufacturing applications. Using the assumption of independent machine component groups, a simple reordering of the listing of machines will decide the order in which the components get listed. The block diagonalization indicating clusters will appear.

Occupancy value for a component can be calculated as:

The route of a component, p_1 consists of a set of machines I_1. All components visiting one or more of these machines can be represented by a set J. Some of these components might be using additional machines, represented by a set I_2. The machines in I_1 and I_2 and components in J represented a machine component sub matrix whose occupancy value is defined as

$$OV_p = \sum_{j \in J} M_j \; / \; (m * n). \qquad\qquad (3.7)$$

where m = number of rows of the sub matrix

n = number of columns of the sub matrix.

ALGORITHM

STEP 1 : Develop an original machine component matrix consisting of ones and zeros. Form another initial machine component matrix identical to the original matrix.

STEP 2 : For each component j in the initial matrix find the total number of machines used (M_j).

STEP 3 : Scan the component list for selecting components with the minimum number of remaining machines (a) If only one component results from the above scan, go to step 5. (b) If there are ties in terms of more than one component having the same minimum number of operations go to step 4.

STEP 4 : Calculate the Occupancy value for each of the components found in step 3. Select the component with the highest occupancy value. Ties for the highest occupancy value can be broken at random.

STEP 5 : Enter the selected component and the machines used into the new matrix.

STEP 6 : Update the initial machine component matrix for all the machines in the following manner. Cross out the rows containing machines selected in step 5. For each component remaining in the initial matrix decrease its M_j value by 1 if a crossed out machine was used by this component.

STEP 7 : Enter any additional components in the new matrix if its M_j value is zero.

STEP 8 : If all machines and components are entered in the new matrix, go to step 9, otherwise, go to step 3.

STEP 9 : Using cell entries from the original machine component matrix as input, compute the new matrix.

The steps of the algorithm are explained in the appendix through an illustration (Appendix A.7).

CHAPTER IV

MEASURES OF COMPARISON

Solution of practical problems without adequate
generalizations and theoretical formulations is useful only in
specific situations. Such results can not usually be applied to
another problem. Group technology has suffered from this malady
for a long time. When researchers and practitioners claim success
with a certain method it is necessary to evaluate the result on
an absolute quantitative scale. The practical use of such a
measure is that subjective individual claims can be compared
objectively. In case of block diagonalization of zero-one
matrices such a quantitative measure can be used as an objective
function to be maximized. Such quantitative measures available in
literature are : (a) grouping efficiency (GE) described by
Chandrasekaran and Rajagopalan [14], (b) Measure of effectiveness
as described by McCormick [48] and (c) Grouping efficiacy as
described by Suresh kumar [18]. First quantitative measures,
grouping efficiency, grouping efficiacy and Measure of
effectiveness are described, followed by issues of bottleneck
machines and exceptional parts.

4.1 GROUPING EFFICIENCY

The concept of grouping efficiency is developed to provide a
quantitative standard on a rational scale for comparing different
solutions to the same problem.

Let $\{D_r\}$, where $r = 1, \ldots k$ be the diagonal sub matrices

obtained by associating the product group P_r with the machine group C_r (in the diagonalized matrix). A perfect grouping implies that

$a_{ij} = 1$ if $a_{ij} \in \{D_r\}$ $r = 1,..k.$ and $a_{ij} = 0$ if $a_{ij} \notin \{D_r\}$.
The diametrically opposite situation, therefore implies that

$a_{ij} = 0$ if $a_{ij} \in \{D_r\}$ $r = 1,...k.$ and $a_{ij} = 1$ if $a_{ij} \notin \{D_r\}$.
The efficiency scale is defined to cover these extreme cases, its value being set at 1 for the former case and zero for the latter.

'Goodness' of grouping depends on two aspects: within group utilization and inter cell movement. From the matrix point of view, the concentration of non-zero elements in the diagonal sub matrices refers to utilization and the presence of such elements outside the diagonal sub matrices represents inter cell movement. A 'better grouping', therefore, means an increase in the utilization, a decrease in the inter cell movement or both.

Grouping efficiency can therefore be considered as a weighted average of two efficiencies η_1 and η_2 defined as follows. The number of non-zero elements in the diagonal blocks e_d is given by

$$e_d = \sum_{r=1}^{k} \sum_{i=M_{r-1}+1}^{M_r} \sum_{j=N_{r-1}+1}^{N_r} a_{ij} \qquad M_0 = 0, N_0 = 0.$$

(4.1)

and the number of non-zero elements outside the diagonal blocks is e_0. Then

$$\eta_1 = e_d \, / \, (\sum_{r=1}^{k} M_r \, N_r)$$

(4.2)

$$\eta_2 = 1 - [\ e_0 \ / \ (\ mn - \sum_{r=1}^{k} M_r \ N_r)\ ].$$ (4.3)

The numerator of the expression for η_1 is the number of non-zero elements in the diagonal blocks and the denominator the total number of elements therein. Similarly, η_2 is the ratio of the number of zeros in the off-diagonal blocks to the total number of elements therein.

The Grouping efficiency η can be expressed as the weighted average of η_1 and η_2.

$$\eta = q\eta_1 + (1 - q)\eta_2$$ (4.4)

where $0 \leq q \leq 1$

η satisfies the basic requirements of non dimensionality, non-negativity and zero to one range. The weighting factor q enables the analyst to alter the emphasis between utilization and inter cell movement, depending on the specific requirements of the given problem.

4.2 LIMITATIONS OF GROUPING EFFICIENCY

The weighted average of η_1 and η_2 gives the analyst enough freedom to decide the relative importance between inter cell movements and voids in diagonal blocks. Chandrasekharan and Rajagopalan contend that a value of q = 0.5 leads to the situation of equal weights to voids and exceptional elements. An analysis of the expression would reveal that this is not true. When large matrices are block diagonalized, for any number of blocks greater than two it can be observed that the first term in expression has a much smaller denominator compared to the second

term, whereas the numerators are more or less of the same order. As the matrix size increases, this disparity between the first and second terms widens and for large and/or sparse matrices the second term becomes less and less effective. In fact, if q and (1 - q) are of the same order, the effect of inter cell movements (exceptional elements) is never reflected in the efficiency values in the case of large and sparse matrices. Therefore to have equal weights for voids and exceptional elements it is necessary to choose a very low value of q. Further, it is desirable to bring the two parts of the expression under a common denominator.

4.2.1 EFFECT OF SPARSENESS OF THE MATRIX

Choice of q would be more rational if it is linked with the size and sparsity of the matrix. As a result q is chosen as the density of an imaginary matrix that has an identical and perfect block diagonal structure as the given matrix.

$$q = \sum_{r=1}^{k} M_r N_r / mn. \qquad (4.5)$$

Placing value of q in the following equation

$$\eta = q\eta_1 + (1 - q)\eta_2.$$

we get

$$\eta = 1 - (e_0 + e_v) / mn. \qquad (4.6)$$

This simplified expression assumes that the perfect diagonal form has 100% efficiency which is reduced by the presence of voids and exceptional elements.

The following characteristics are important in the case of the modified expression for η after choosing q as discussed earlier.

The main difference between this expression and the earlier expression for efficiency is that the present expression has the size of the matrix (mn) as its denominator, whereas in earlier expression η_1 and η_2 have two different denominators. It is also interesting to note that it is possible to assign relative weights for voids and exceptions in the present expression also, because it is in the present form that voids and exceptional elements have, an reality, equal weights.

A major defect of grouping efficiency in both forms is its low discriminating power. It is obvious that $(e_0 + e_v)$ is much less than the size of the matrix 'mn'. Thus, for large matrices the term $(e_0 + e_v) / mn$ will take values of the order of 10 which leads to an efficiency close to one.

Another serious defect lies in the definition of zero efficiency. Zero has been defined as the most degenerate case where $e_0 + e_v = mn$. This implies that the diagonal blocks are full of zeros and off-diagonal blocks are full of ones. It can be seen that this definition is un-realistic from physical point of view.

4.2.2 EFFECT OF SIZE OF THE MATRIX

By accepting 'e' as denominator of the efficiency function it will have the form

$$\eta = 1 - (e_0 + e_v) / e.$$

$$= (e - e_0 - e_v) / e.$$

$$= (e_d - e_v) / e. \tag{4.7}$$

The main drawbacks of this function are as follows

(1) Zero point is not properly defined. η becomes zero when $e_d = e_v$ i.e.

$$e_d = e_v = 1/2 \{ \sum_{r = 1}^{k} M_r N_r \}. \tag{4.8}$$

This means that the zero efficiency is dictated by the fact that there are 50% voids in the diagonal blocks irrespective of the number of exceptional elements present.

(2) This function takes negative values when $e_v < e_d$.

4.3 GROUPING EFFICIACY

For a large denominator the discriminating power of the efficiency function is poor and for a small denominator the function itself may become negative.

The meaningful denominator between 'e' and 'mn' is the 'operational zone' of the matrix. The operational zone of block diagonal form consists of the diagonal blocks usually consist of both ones and voids, and the exceptional elements consist of only ones.

Operational zone = Voids + Ones in the diagonal blocks + Ones in off-diagonal blocks.

$$= e_v + e_d + e_0.$$

$$= e_v + e. \tag{4.9}$$

Thus expression can be modified as

$$\eta = 1 - (e_v + e_0) \,/\, (e_v + e)$$

$$= (e_v + e - e_v - e_0) \,/\, (e_v + e)$$

$$= (e - e_0) \,/\, (e_v + e). \tag{4.10}$$

This is grouping efficiacy (GEF).

$$GEF = (1 - \Psi)/(1 + \Phi) \tag{4.11}$$

$$\Psi = \frac{\text{no. of exceptional elements}}{\text{Total no. of operations}}$$

$$\Phi = \frac{\text{no. of voids in diagonal blocks}}{\text{Total no. of operations}}$$

4.4 MEASURE OF EFFECTIVENESS

The measure of effectiveness (Measure of effectiveness) introduced in McCormick[48] is used to measure the quality of a solution. The measure is defined as follows:

$$\text{Measure of effectiveness} = \frac{1}{2} \sum_1^m \sum_1^n a_{i,j} * (a_{i,j-1} + a_{i+1,j} + a_{i,j+1} + a_{i-1,j}). \tag{4.12}$$

where $a_{i,j}$ is the (i,j)th element of the machine and part incidence matrix.

The Measure of effectiveness is specifically designed for data array clustering. This measure however is sensitive to the relative positioning of rows and columns of a matrix. The same grouping solutions with different arrangement of rows and columns may give different values of the measure.

4.5 EXCEPTIONAL PARTS AND BOTTLENECK MACHINES

Most of the cell formation techniques employ part routeing information in the form of machine component 0-1 incidence matrix. A one as an entry in the matrix indicates that the corresponding machine is required to process corresponding part and a zero indicates otherwise. The matrix is rearranged to identify the machine cells and part families.

Clustering of a binary incidence matrix may result in the following two categories of clusters:

(1) Mutually separable clusters.

(2) Partially separable clusters.

```
0      1  3  2  4  5
     ┌─────────────────┐
2    │ 1  1            │
     │                 │
4    │ 1  1            │
     │                 │
1    │       1  1  1   │
     │                 │
3    │       1  1      │
     └─────────────────┘
```

Figure 4.1 : Mutually Seperable Cluster

The above figure is Mutually separable cluster whereas the figure below is a partially separable cluster.

```
0      1  2  3  4  5
     ┌─────────────────┐
1    │ 1  1  1         │
     │                 │
2    │ 1  1            │
     │                 │
3    │       1  1  1   │
     │                 │
4    │       1  1      │
     └─────────────────┘
```

Figure 4.2 : Partially Seperable Cluster

There may be instances where a part may appear in more than

one part family indicating that the processing of the part cannot be done in a single cell and thus an inter cell transfer would be required such parts are termed as exceptional parts.

Matrix 2 cannot be separated into disjoint clusters because of part 5, which is to be machined in two cells, MC-1 and MC-2. Removing part 5 from matrix results in the decomposition of matrix into part families PF-1 = {1,2} and PF-2 = {3,4}. The two clusters are called the partially seperable clusters.

The presence of exceptional parts in the final solution of a cell formation problems may be due to the weakness of the algorithm or, in certain problem situations, it may be inevitable to avoid exceptional parts.

Following actions to deal with exceptional parts can be taken.

(1) It can be machined in one machine cell and transferred to the other machine cell by a material handling carrier.

(2) It can be machined in a functional facility.

(3) It can be sub-contracted.

A bottleneck machine is a machine that does not allow decomposition of a machine part incidence matrix into sub matrices.

Machine 3 in the figure 4.3 does not permit decomposition of that matrix into two machine cells and two part families.

0	1	2	3	4	5	6
1	1	1				
2	1	1				
3	1	1	1		1	1
4			1	1	1	1
5			1		1	1

Figure 4.3 : Effect of Bottleneck Machine

A way to decompose matrix into two mutually separable sub matrices is to purchase an additional copy of machine 3. This will yield following figure.

0	1	2	3	4	5	6	
1	1	1					
2	1	1					MC-1
3	1	1					
3			1		1	1	
4			1	1	1	1	MC-2
5			1		1	1	

Figure 4.4 : On Duplicating Bottleneck Machine

CHAPTER V

IMPLEMENTATION, EXAMPLES, OBSERVATIONS AND EXPLANATIONS

In order to compare the algorithms of cell formation discussed in chapter III several representative illustrations were choosen. Algorithms were coded using PASCAL and the system was implemented on HP 9000. Grouping efficiancy, grouping efficiacy, measure of effectiveness, number of bottleneck machines and number of exceptional parts were computed using the representative examples. Time for computation was recorded on HP system. Four representative examples which are selected for the discussions are given in section 4.1. Grouping efficiancy, grouping efficiacy, measure of effectiveness, improvement in measure of effectiveness, number of bottleneck machines and exceptional parts, and time are shown with each figure and at the same time are summarised in tables 1 to 4.

5.1 REPRESENTATIVE EXAMPLES

```
    0 0 0 0 0 0 0 0 0 1
    0 1 2 3 4 5 6 7 8 9 0
 1  0 0 1 1 0 1 0 0 0 0
 2  1 0 0 0 0 1 0 0 1
 3  0 1 0 0 1 0 0 1 0 0
 4  0 0 0 1 0 1 0 0 1 0
 5  0 1 0 0 1 0 0 1 0 0
 6  0 0 1 0 0 1 0 0 1 0
 6  0 0 1 0 0 1 0 0 1 0
 7  0 0 0 0 0 1 0 0 1
 8  0 1 0 0 1 0 0 1 0 0
 9  0 0 1 1 0 1 0 0 1 0
10  1 0 0 0 0 0 1 0 0 1
11  1 0 0 0 0 0 1 0 0 1
12  1 0 0 0 0 0 1 0 0 1
13  0 1 0 0 1 0 0 1 0 0
14  0 0 1 1 0 1 0 0 1 0
15  0 1 0 0 1 0 0 1 0 0
```

Figure 5.1 : Example 1

```
┌                             ┐
│  0  0 1 0 0 0 0 0 0 0 0      │
│  0  7 0 1 2 5 8 6 3 4 9      │
│  2  1 1 1 0 0 0 0 0 0 0      │
│ 10  1 1 1 0 0 0 0 0 0 0      │
│ 11  1 1 1 0 0 0 0 0 0 0      │
│ 12  1 1 1 0 0 0 0 0 0 0      │
│  7  1 1 0 0 0 0 0 0 0 0      │
│  3  0 0 0 1 1 1 0 0 0 0      │
│  5  0 0 0 1 1 1 0 0 0 0      │
│  8  0 0 0 1 1 1 0 0 0 0      │
│ 13  0 0 0 1 1 1 0 0 0 0      │
│ 15  0 0 0 1 1 1 0 0 0 0      │
│  9  0 0 0 0 0 0 1 1 1 1      │
│ 14  0 0 0 0 0 0 1 1 1 1      │
│  1  0 0 0 0 0 0 1 1 1 0      │
│  6  0 0 0 0 0 0 1 1 0 1      │
│  4  0 0 0 0 0 0 1 0 1 1      │
└                             ┘
```

GE = 0.96   BM = 0 EP = 0   ME = 63   IME = 54   GEF = 0.92

Figure 5.2 : Output by King's ROC Method

```
┌                             ┐
│  0  0 0 0 0 0 0 0 0 1 0      │
│  0  2 5 8 6 3 4 9 7 0 1      │
│  3  1 1 1 0 0 0 0 0 0 0      │
│  5  1 1 1 0 0 0 0 0 0 0      │
│  8  1 1 1 0 0 0 0 0 0 0      │
│ 13  1 1 1 0 0 0 0 0 0 0      │
│ 15  1 1 1 0 0 0 0 0 0 0      │
│  9  0 0 0 1 1 1 1 0 0 0      │
│ 14  0 0 0 1 1 1 1 0 0 0      │
│  1  0 0 0 1 1 1 0 0 0 0      │
│  6  0 0 0 1 1 0 1 0 0 0      │
│  4  0 0 0 1 0 1 1 0 0 0      │
│  2  0 0 0 0 0 0 0 1 1 1      │
│ 10  0 0 0 0 0 0 0 1 1 1      │
│ 11  0 0 0 0 0 0 0 1 1 1      │
│ 12  0 0 0 0 0 0 0 1 1 1      │
│  7  0 0 0 0 0 0 0 1 1 0      │
└                             ┘
```

GE = 0.96 BM = 0 EP = 0 ME = 63   IME = 54   GEF = 0.92

Figure 5.3 : Output by Graham's LWA Method

```
 0  0 0 0 0 0 0 1 0 0 0 0
 0  2 5 6 7 8 0 1 3 4 9
 7  0 0 0 1 0 1 0 0 0 0
 1  0 0 1 0 0 0 0 1 1 0
 2  0 0 0 1 0 1 1 0 0 0
 3  1 1 0 0 1 0 0 0 0 0
 4  0 0 1 0 0 0 0 0 1 1
 5  1 1 0 0 1 0 0 0 0 0
 6  0 0 1 0 0 0 0 1 0 1
 8  1 1 0 0 1 0 0 0 0 0
10  0 0 0 1 0 1 1 0 0 0
11  0 0 0 1 0 1 1 0 0 0
12  0 0 0 1 0 1 1 0 0 0
13  1 1 0 0 1 0 0 0 0 0
15  1 1 0 0 1 0 0 0 0 0
 9  0 0 1 0,0 0 0 1 1 1
14  0 0 1 0 0 0 0 1 1 1
```

GE = 0.66   BM = 0   EP = 10   ME = 28   IME = 19   GEF = 0.371

Figure 5.4 : Output by DCA Method

```
 0  0 0 0 0 0 0 1 0 0 0
 0  3 4 6 9 1 7 0 2 5 8
 1  1 1 1 0 0 0 0 0 0 0
 4  0 1 1 1 0 0 0 0 0 0
 6  1 0 1 1 0 0 0 0 0 0
 9  1 1 1 1 0 0 0 0 0 0
14  1 1 1 1 0 0 0 0 0 0
 2  0 0 0 0 1 1 1 0 0 0
 7  0 0 0 0 0 1 1 0 0 0
10  0 0 0 0 1 1 1 0 0 0
11  0 0 0 0 1 1 1 0 0 0
12  0 0 0 0 1 1 1 0 0 0
 3  0 0 0 0 0 0 0 1 1 1
 5  0 0 0 0 0 0 0 1 1 1
 8  0 0 0 0 0 0 0 1 1 1
13  0 0 0 0 0 0 0 1 1 1
15  0 0 0 0 0 0 0 1 1 1
```

GE = 0.96   BM = 0   EP = 0   ME = 63   IME = 54   GEF = 0.92

Figure 5.5 : Output by CIA Method

```
1 1 1 0 0 0 0 0 0 0
1 1 1 0 0 0 0 0 0 0
1 1 1 0 0 0 0 0 0 0
1 1 1 0 0 0 0 0 0 0
1 1 0 0 0 0 0 0 0 0
0 0 0 1 1 1 0 0 0 0
0 0 0 1 1 1 0 0 0 0
0 0 0 1 1 1 0 0 0 0
0 0 0 1 1 1 0 0 0 0
0 0 0 1 1 1 0 0 0 0
0 0 0 0 0 0 1 1 1 1
0 0 0 0 0 0 1 1 1 1
0 0 0 0 0 0 1 1 1 0
0 0 0 0 0 0 1 1 0 1
0 0 0 0 0 0 1 0 1 1
```

GE = 0.96   BM = 0   EP = 0   ME = 63   IME = 54   GEF = 0.92

Figure 5.6 : Output by ROC2 Method

```
   0 0 0 0 0 0 1 0 0 0 0
   0 2 5 8 1 7 0 3 6 4 9
   1 0 0 0 0 0 0 1 1 1 0
   7 0 0 0 0 1 1 0 0 0 0
  10 0 0 0 1 1 1 0 0 0 0
   2 0 0 0 1 1 1 0 0 0 0
   5 1 1 1 0 0 0 0 0 0 0
   8 1 1 1 0 0 0 0 0 0 0
   3 1 1 1 0 0 0 0 0 0 0
   6 0 0 0 0 0 0 1 1 0 1
   4 0 0 0 0 0 0 0 1 1 1
   9 0 0 0 0 0 0 1 1 1 1
  11 0 0 0 1 1 1 0 0 0 0
  12 0 0 0 1 1 1 0 0 0 0
  13 1 1 1 0 0 0 0 0 0 0
  14 0 0 0 0 0 0 1 1 1 1
  15 1 1 1 0 0 0 0 0 0 0
```

GE = 0.61   BM = 5   EP = 0   ME = 49   IME = 40   GEF = 0.348

Figure 5.7 : Output by BEA Method

```
 0  0 0 1 0 0 0 0 0 0 0
 0  1 7 0 3 4 9 6 2 8 5
 2  1 1 1 0 0 0 0 0 0 0
10  1 1 1 0 0 0 0 0 0 0
11  1 1 1 0 0 0 0 0 0 0
12  1 1 1 0 0 0 0 0 0 0
 7  0 1 1 0 0 0 0 0 0 0
 1  0 0 0 1 1 0 1 0 0 0
 6  0 0 0 1 0 1 1 0 0 0
 9  0 0 0 1 1 1 1 0 0 0
14  0 0 0 1 1 1 1 0 0 0
 4  0 0 0 0 1 1 1 0 0 0
 3  0 0 0 0 0 0 0 1 1 1
 5  0 0 0 0 0 0 0 1 1 1
 8  0 0 0 0 0 0 0 1 1 1
13  0 0 0 0 0 0 0 1 1 1
15  0 0 0 0 0 0 0 1 1 1
```

GE = 0.96   BM = 0   EP = 0   ME = 64   IME = 55   GEF = 0.92

Figure 5.8 : Output by OVM Method

```
    0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 2
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
 1  1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 2  0 1 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0
 3  0 1 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0
 4  1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 5  0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1 0 1
 6  1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 7  0 0 0 0 0 1 0 0 1 0 1 1 0 0 0 1 0 0 1 0
 8  0 0 0 0 0 1 0 0 1 0 1 1 0 0 0 1 0 0 1 0
 9  0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1 0 1
10  0 0 0 0 0 1 0 0 1 0 1 1 0 0 0 1 0 0 1 0
```

Figure 5.9 : Example 2

```
 0 0 0 0 0 0 0 0 1 0 0 1 1 1 1 1 1 1 1 2
 0 1 4 7 2 3 5 8 0 6 9 1 2 6 9 3 4 5 7 8 0
 1  1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 4  1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 6  1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 2  0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
 3  0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
 7  0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0
 8  0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0
10  0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0
 5  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
 9  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
```

GE = 1   BM = 0   EP = 0   ME = 68   IME = 46   GEF = 1

Figure 5.10 : Output by ROC Method

```
 0 0 0 1 1 1 1 0 0 0 0 1 0 0 0 1 1 1 1 1 2
 0 6 9 1 2 6 9 2 3 5 8 0 1 4 7 3 4 5 7 8 0
 7  1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 8  1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
10  1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 2  0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0
 3  0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0
 1  0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0
 4  0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0
 6  0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0
 5  0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
 9  0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
```

GE = 1   BM = 0   EP = 0   ME = 68   IME = 46   GEF = 1

Figure 5.11 : Output by LWA Method

```
 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 1 1 2
 0 2 4 6 7 9 1 2 6 9 2 3 5 8 0 3 4 5 7 8 0
 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 4 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 6 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 2 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0
 3 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0
 5 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
 7 0 0 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
 8 0 0 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
10 0 0 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
```

GE = 0.799   BM = 0   EP = 1   ME = 60   IME = 38   GEF = 0.6

Figure 5.12 : Output by DCA Method

```
 0 1 1 1 1 1 2 0 0 1 1 1 1 0 0 0 0 1 0 0 0
 0 3 4 5 7 8 0 6 9 1 2 6 9 2 3 5 8 0 1 4 7
 5 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 9 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 7 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0
 8 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0
10 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0
 2 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0
 3 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0
 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1
 4 0 0 0 0 0 G 0 0 0 0 0 0 0 0 0 0 1 1 1
 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1
```

GE = 1   BM = 0   EP = 0   ME = 68   IME = 46   GEF = 1

Figure 5.13 : Output by CIA Method

```
1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
```

GE = 1   BM = 0   EP = 0   ME = 68   IME = 46   GEF = 1

Figure 5.14 : Output by ROC2 Method

```
    0 0 0 0 0 1 0 0 0 0 0 1 1 1 1 1 1 1 1 1 2
    0 2 3 5 8 0 1 4 7 6 9 1 2 6 9 3 4 5 7 8 0
 1  0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 4  0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 7  0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0
 2  1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 3  1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 5  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
 8  0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0
10  0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0
 6  0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 9  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
```

GE = 0.81   BM = 0   EP = 0   ME = 62   IME = 40   GEF = 0.62

Figure 5.15 : Output by BEA Method

```
 0  0 1 0 0 0 1 2 1 1 1 1 0 0 0 0 1 1 1 1 0
 0  2 0 8 5 3 3 0 8 7 5 4 1 7 4 6 9 6 2 1 9
 2  1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 3  1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 5  0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
 9  0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
 1  0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0
 4  0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0
 6  0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0
 7  0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
 8  0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
10  0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
```

GE = 1   BM = 0   EP = 0   ME = 68   IME = 46   GEF = 1

Figure 5.16 : Output by OVM Method

```
    0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
 1  0 0 0 0 0 1 1 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 2  0 0 1 1 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 3  0 0 1 1 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  1  0  0  0
 4  1 1 0 0 0 0 0 0 0 0  0  0  0  0  0  1  0  1  1  0  0  1  0
 5  1 0 0 0 0 0 0 0 0 0  0  0  0  0  0  1  0  0  1  0  0  1  0
 6  0 0 0 0 0 0 0 0 1 1  1  1  0  1  1  1  0  0  0  0  1  0  0
 7  1 1 0 0 0 1 0 0 0 0  0  0  0  0  0  1  0  0  1  0  0  0  0
 8  0 0 0 0 1 0 0 0 1 1  0  1  0  1  1  1  0  0  0  0  1  0  0
 9  0 0 0 0 1 0 0 0 1 0  0  1  1  0  1  0  0  0  0  0  0  0  0
10  0 1 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
11  0 0 1 1 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  1  0  0  1
12  0 0 0 0 0 0 1 1 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
13  0 0 0 0 0 1 1 1 0 0  0  0  0  0  0  1  0  0  0  0  1  0
14  0 0 0 0 0 0 0 0 1 0  1  0  1  0  1  0  0  0  0  0  0  0
```

Figure 5.17 : Example 3

```
 0 11720 2 6231924 7 818 3 421 91512 5101416221311
 7 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 4 1 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 5 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
10 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0·0 0 0 0 0
13 0 0 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
11 0 0 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0
12 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 3 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0
 2 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0
 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0
 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1
14 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1
```

GE = 0.75 BM = 1 EP = 1 ME = 58 IME = 26 GEF = 0.5

Figure 5.18 : Output by ROC Method

```
 0 91512 117201014162223 5 211 613 3 4 72119 82418
 8 1 1 1 0 0 0 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0
 6 1 1 1 0 0 0 1 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0
 9 1 1 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0
 4 0 0 0 1 1 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0
 7 0 0 0 1 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0
 5 0 0 0 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
14 1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0
13 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 1 0 0 1
11 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0
 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0
 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0
 2 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0
10 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0
12 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0
```

GE = 0.63 BM = 1 EP = 3 ME = 42 IME = 10 GEF = 0.3

Figure 5.19 : Output by LWA Method

```
 0 915 1 2 3 4 6 712172023 5 810111314162122241819
 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 2 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
10 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
12 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
 3 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
 5 0 0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
11 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0
14 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0
 7 0 0 1 1 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 9 1 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0
13 0 0 0 0 0 0 1 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0
 4 0 0 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1
 6 1 1 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 1 1 0 1 0 0 0
 8 1 1 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 1 1 0 1 0 0 0
```

GE = 0.6 BM = 1 EP = 14 ME = 28 IME = 0 GEF = 0.24

Figure 5.20 : Output by DCA Method

```
 0 5 910111213141516 22 1 2 3 4 6 7 817181920212324
 6 0 1 1 1 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 8 1 1 1 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 9 1 1 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
14 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
 2 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
 3 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0
 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 1 1 0 1 0
 5 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 1 0
 7 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 1 0 0 0
10 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1
11 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 1
12 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0
13 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 0 1 0
```

GE = 0.67 BM = 0 EP = 0 ME = 50 IME = 18 GEF = 0.33

Figure 5.21 : Output by CIA Method

```
1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1
```

GE = 0.75 BM = 1 EP = 1 ME = 58 IME = 26 GEF = 0.5

Figure 5.22 : Output by ROC2 Method

```
   0191823 8 7 12017 6 22421 3 413 5 922161514121110
 4 1 0 1 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
13 0 1 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 5 0 0 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
12 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 7 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
10 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
11 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0
 3 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0
 2 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
 9 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 1 0 0
14 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 0
 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 1
 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
```

GE = 0.74 BM = 0 EP = 1 ME = 57 IME = 25 GEF = 0.48

Figure 5.23 : Output by OVM Method

```
   0 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293 0
 1 0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 1 1 0 0 1 0 0 0 1 0
 2 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 1 1 1 0 0 1 1 1 0 0 0 0
 3 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0
 4 0 0 1 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1
 5 0 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0
 6 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1 1 1 1
 7 0 0 0 0 0 0 1 0 0 0 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 8 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0
 9 0 1 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 1 0 1
10 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
11 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 1
12 1 0 1 1 0 0 1 0 0 0 1 0 1 0 1 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0
13 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 1 0 0 0 0 1 0 1 1 1 0 1 0
14 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 1 0 0 0
15 1 0 1 1 0 0 1 0 0 0 1 0 1 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0
16 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1
```

Figure 5.24 : Example 4

```
   016 11113 3 4 71419181521 6122230 92824 823262725171029 220 5
15 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
12 1 1 1 1 1 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 8 1 1 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
10 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 5 1 0 1 1 1 1 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 7 1 0 1 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
11 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
 3 0 0 1 1 0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 4 0 0 0 1 1 0 1 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
13 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 1 1 1 1 1 1 0 0 0
 2 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 1 1 1 1 0 0 1 1 1
 6 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 1 1 1 0 0 1 1 0 0 0
 9 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 1 0 1 0 1 0 0
 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 1 1 0 0 0 1 1 0 1 1
16 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 1 1 1 0 1 0 1 0
14 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 0
```

GE = 0.65   BM = 1   EP = 1   ME = 113   IME = 57   GEF = 0.37

Figure 5.25 : Output by ROC Method

```
    01322111516 31223 7 42630252910 12714 91821172028 824 5 219 6
12 1 0 1 1 1 1 0 0 1 1 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0
 5 1 1 1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
15 1 0 1 0 1 1 0 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0
 4 1 1 0 0 0 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
13 0 0 0 1 0 0 0 1 0 0 1 0 1 1 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0
 3 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 1 0 1 0 0 0 0 1 1 0 0 1 0 0 1 1 0 0 0 0 0 0 1 0 1 0 1 0 0 0
16 0 0 0 0 0 0 0 1 0 0 1 1 1 1 0 0 1 0 0 0 0 1 1 1 1 0 0 0 0
 9 0 1 0 0 0 0 1 0 0 0 0 1 1 0 1 0 0 0 1 0 0 0 0 1 0 1 0 1 0 0
 2 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 1 0 1 1 1 0 0 0 1 1 0 0
 6 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0
 7 0 0 1 1 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
11 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 0
10 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 8 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1
14 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0
```

GE = 0.58   BM = 1   EP = 14   ME = 80   IME = 24   GEF = 0.29

Figure 5.26 : Output by LWA Method

```
    01611121522313252630 1 3 7 8 91020272829 41417182124 2 5 619
10 1 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
14 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
 7 1 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
 8 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0
 3 0 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
 4 0 0 0 0 1 1 1 0 0 1 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0
11 1 0 1 0 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0
 5 1 1 1 1 1 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
 6 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 1 0 1 1 1 0 0 0 0 1 0 0 0 0 0
 1 0 0 1 0 1 1 0 0 1 0 0 0 0 1 0 1 1 0 0 1 0 0 0 0 0 0 0 1 0 0
 9 0 0 1 0 1 0 0 1 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 1 1 0 0 0
13 0 0 0 1 0 1 0 1 1 0 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 0 0 0
15 1 1 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1
 2 0 0 0 0 1 0 0 1 1 0 0 0 0 0 1 0 1 1 0 0 0 0 1 0 1 0 1 1 0 0
12 1 1 0 1 0 0 1 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0
16 0 0 0 0 0 1 0 1 1 1 0 0 0 1 0 0 1 1 1 1 0 0 1 0 0 1 0 0 0
```

GE = 0.56   BM = 0   EP = 22   ME = 59   IME = 3   GEF = 0.262

Figure 5.27 : Output by DCA Method

OUTPUT BY KING'S ROC2 METHOD

```
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 1 1 1 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 1 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 1 1 1 1 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 1 1 1 0 0 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 1 0 1 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 1 1 0 0 0 1 1 0 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 1 1 0 1 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 0
```

GE = 0.68   BM = 2   EP = 6   ME = 113   IME = 57   GEF = 0.40

Figure 5.28 : Output by ROC2 Method

```
   0 2 9 25 17 26 23 29 10 12 22 13 3 3 11 16 1 11 4 15 18 6 5 20 8 24 28 30 27 21 19
 1 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0
16 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0
11 0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1 1 1 1 0 0 0
13 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0
 3 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0
 4 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0
 7 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0
14 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0
15 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1
12 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 1 1 0 0 0 0 0 0 1 0
 8 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0
 5 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
 2 1 1 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 1 0
 9 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0
10 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0
 6 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0
```

GE = 0.57   BM = 7   EP = 14   IME = 55   GEF = 0.276

Figure 5.29 : Output by BEA Method

```
    0 619 4 118111516 31413 72412 81030392823 227262522212017 9 5
 8  1 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
15  0 1 1 1 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 5  0 0 1 0 0 1 1 1 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
12  0 0 1 1 1 1 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
10  0 0 0 1 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 3  0 0 0 0 1 1 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
 7  0 0 0 0 0 1 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
13  0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 1 0 1 1 1 0 0 0 1 1 0
11  0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0
 4  0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0
 9  0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0 1 0 1 0 0 1 1 0 0 0 1 0
16  0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 1 1 1 0 1 1 1 0 0 1 1 0 0
 1  0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 1 0 1 0 0 1 0 1 0 1 0 0 1
14  0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0
 6  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 1 0 0 1 0 0 0 0
 2  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1
```

GE = 0.70   BM = 1   EP = 4   ME = 98   IME = 42   GEF = 0.42

Figure 5.30 : Output by OVM Method

TABLE 1 : COMPARISON OF VARIOUS ALGORITHMS

PROBLEM SIZE		ROC	LWA	DCA	CIA	ROC2	BEA	OVM
	NO. OF BM[1]	0	0	0	0	0	*	0
	NO. OF EP[2]	0	0	10	0	0	*	0
15 MCS 10 PRTS	GE[3]	96%	96%	66%	96%	96%	62%	96%
	ME[4]	63	63	28	63	63	49	64
	IME[5]	54	54	19	54	54	40	55
	GEF[6]	92%	92%	37%	92%	92%	35%	92%
	TIME	0.15s	0.12s	0.07s	0.09s	0.12s	0.06s	0.11s

TABLE 2 : COMPARISON OF VARIOUS ALGORITHMS

PROBLEM SIZE		ROC	LWA	DCA	CIA	ROC2	BEA	OVM
	NO. OF BM	1	1	1	0	1	*	0
	NO. OF EP	1	3	14	0	1	*	1
14 MCS 24 PRTS	GE	75%	63%	59%	66%	75%	*	74%
	ME	58	42	28	50	58	*	57
	IME	26	10	-4	18	26	*	25
	GEF	49%	30%	24%	33%	50%	16%	48%
	TIME	0.25s	0.21s	0.14s	0.12s	0.28s	0.15s	0.3s

1 BM = Bottleneck machines
2 EP = Exceptional parts
3 GE = Grouping efficiancy
4 ME = Measure of effectiveness
5 IME = Improvement in measure of effectiveness
6 GEF = Grouping efficiacy

TABLE 3 : COMPARISON OF VARIOUS ALGORITHMS

PROBLEM SIZE		ROC	LWA	DCA	CIA	ROC2	BEA	OVM
	NO. OF BM	0	0	0	0	0	0	0
	NO. OF EP	0	0	1	0	0	0	0
10 MCS 20 PRTS	GE	100%	100%	80%	100%	100%	81%	100%
	ME	68	68	60	68	68	53	68
	IME	46	46	38	46	46	31	46
	GEF	100%	100%	60%	100%	100%	62%	100
	TIME	0.9s	0.13s	0.05s	0.07s	0.05s	0.13s	0.16s

TABLE 4 : COMPARISON OF VARIOUS ALGORITHMS

PROBLEM SIZE		ROC	LWA	DCA	CIA	ROC2	BEA	OVM
	NO. OF BM	1	1	0	*	2	7	1
	NO. OF EP	9	14	22	*	6	14	4
16 MCS 30 PRTS	GE	66%	58%	56%	*	68%	57%	70
	ME	113	80	59	56	113	111	98
	IME	57	24	3	0	57	55	42
	GEF	37%	30%	26%	25%	40%	28%	42
	TIME	0.33s	0.20s	0.16s	0.14s	0.25s	0.25s	0.5

*   Calculation not possible

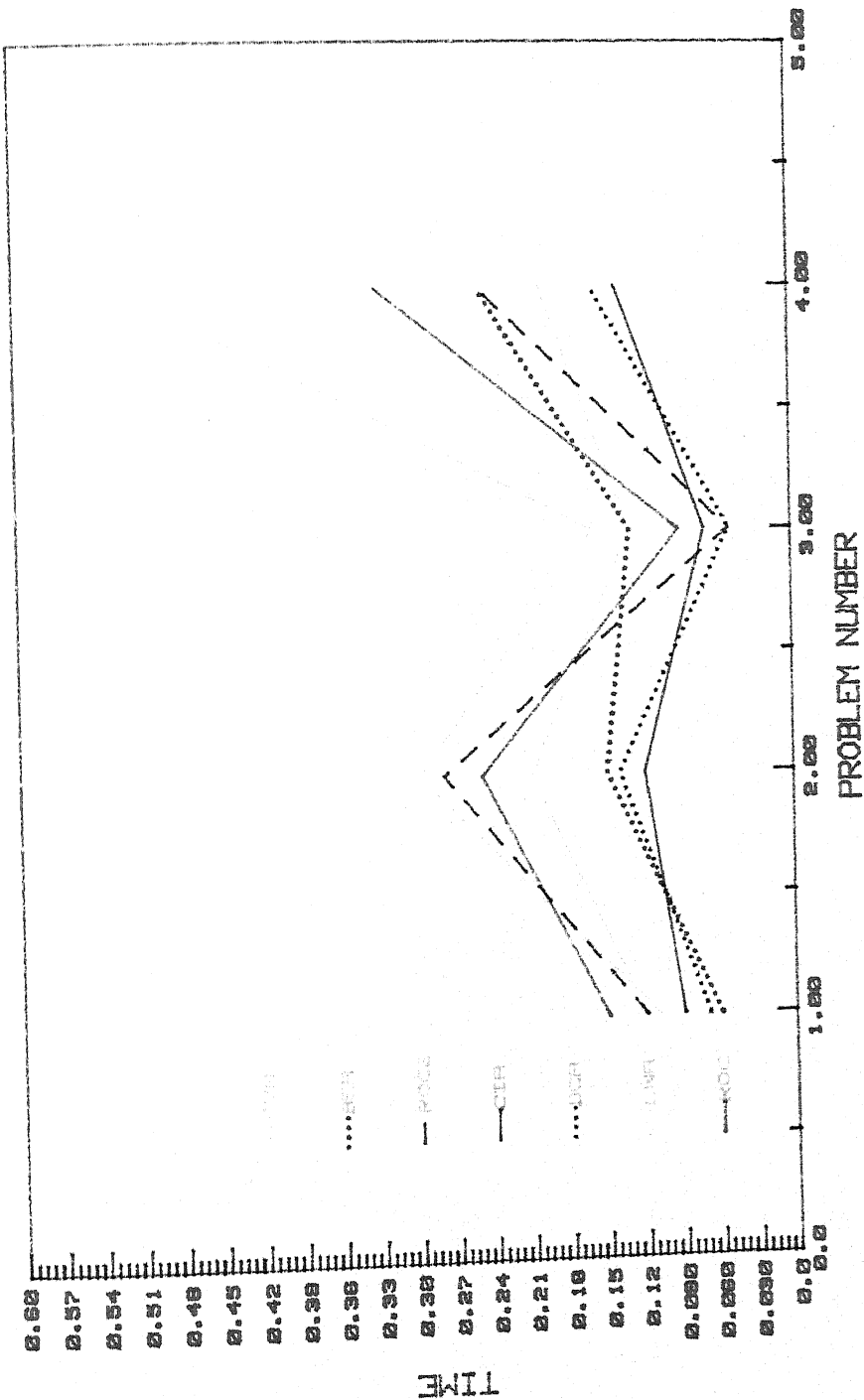COMPARISON OF GROUPING EFFICIENCY

Figure Number 5.31

PROBLEM NUMBER

GROUPING EFFICIENCY

OVM
SEA
ROC2
CIR
XCR
WA
ROC
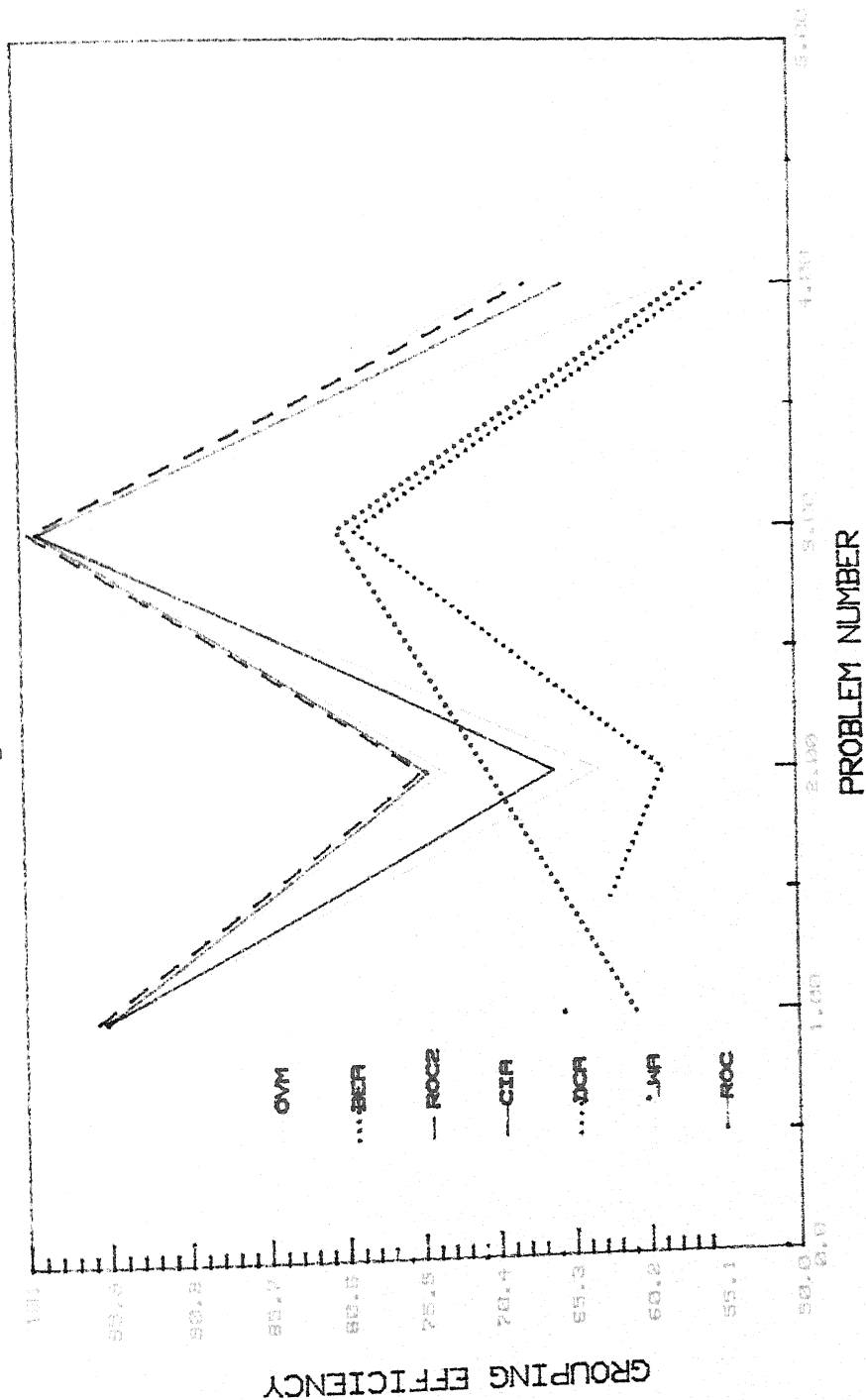
COMPARISON OF MEASURE OF EFFECTIVENESS

Figure No. 5.32

ME

Problem No.
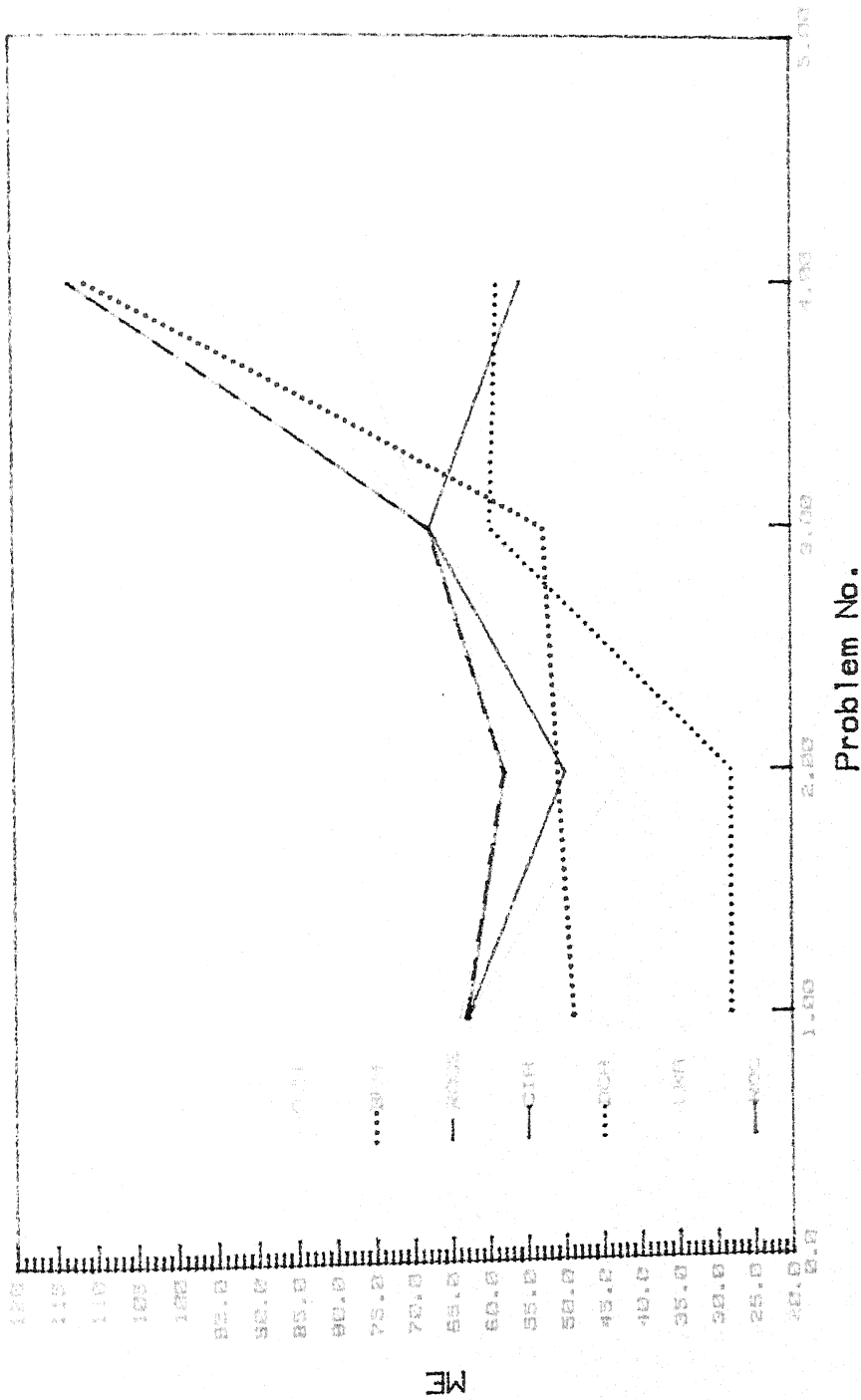
COMPARISON OF TIME

Figure No. S.33
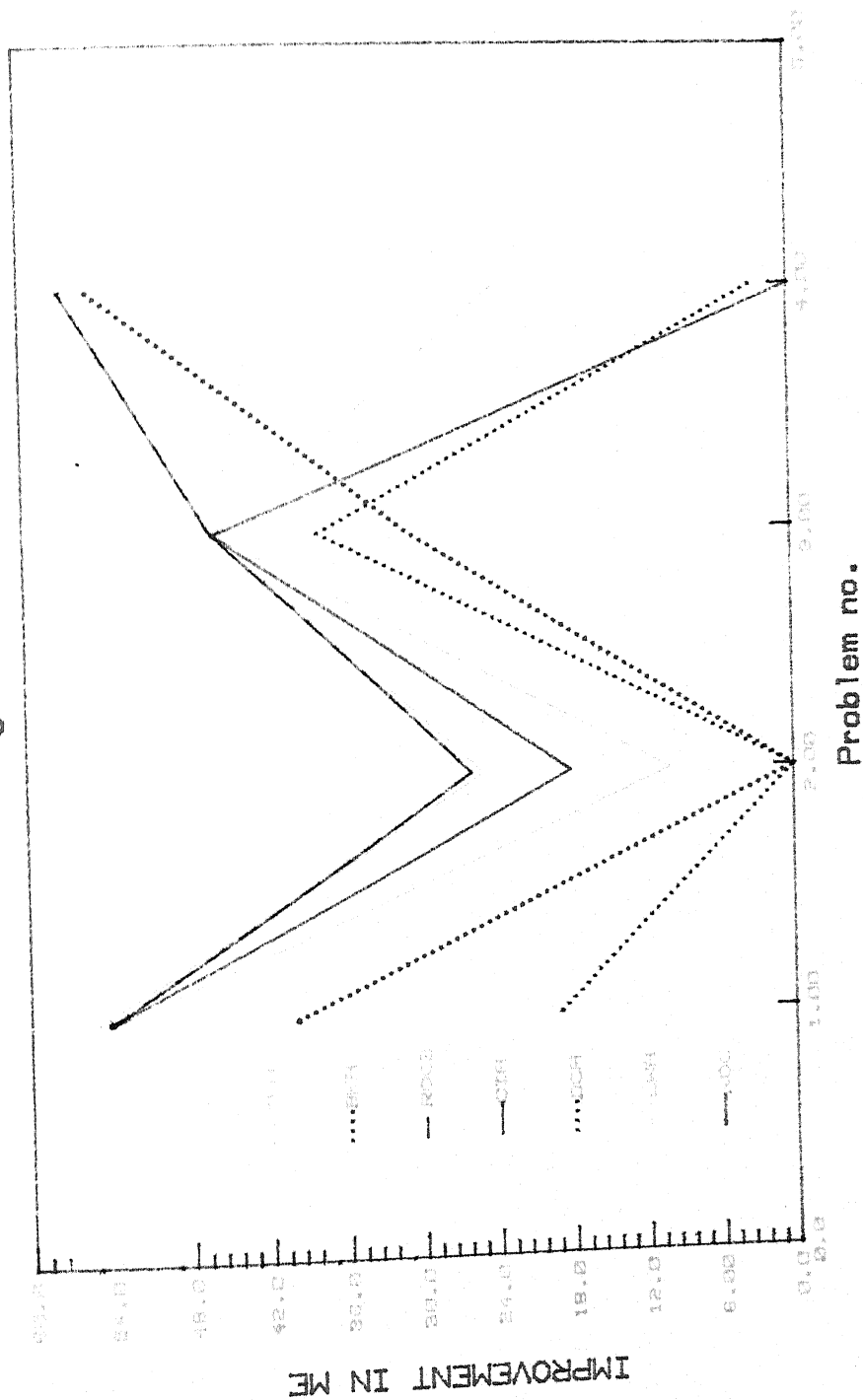

PROBLEM NUMBER

TIME

COMPARISON OF IMPROVEMENT IN ME

Figure No. S.34

Graphstar
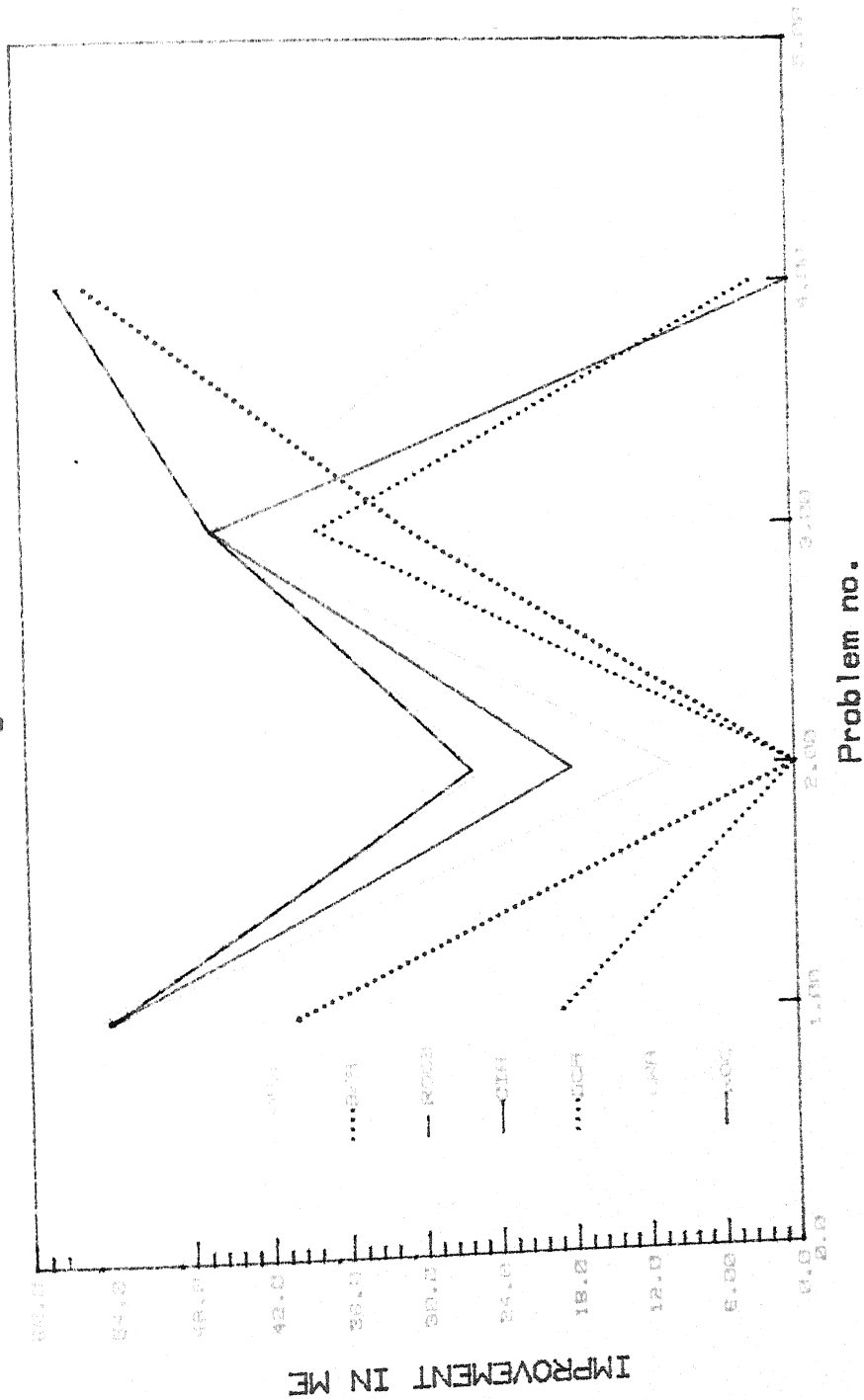
COMPARISON OF IMPROVEMENT IN ME

Figure No. S.34

Problem no.

IMPROVEMENT IN ME

Graphstar
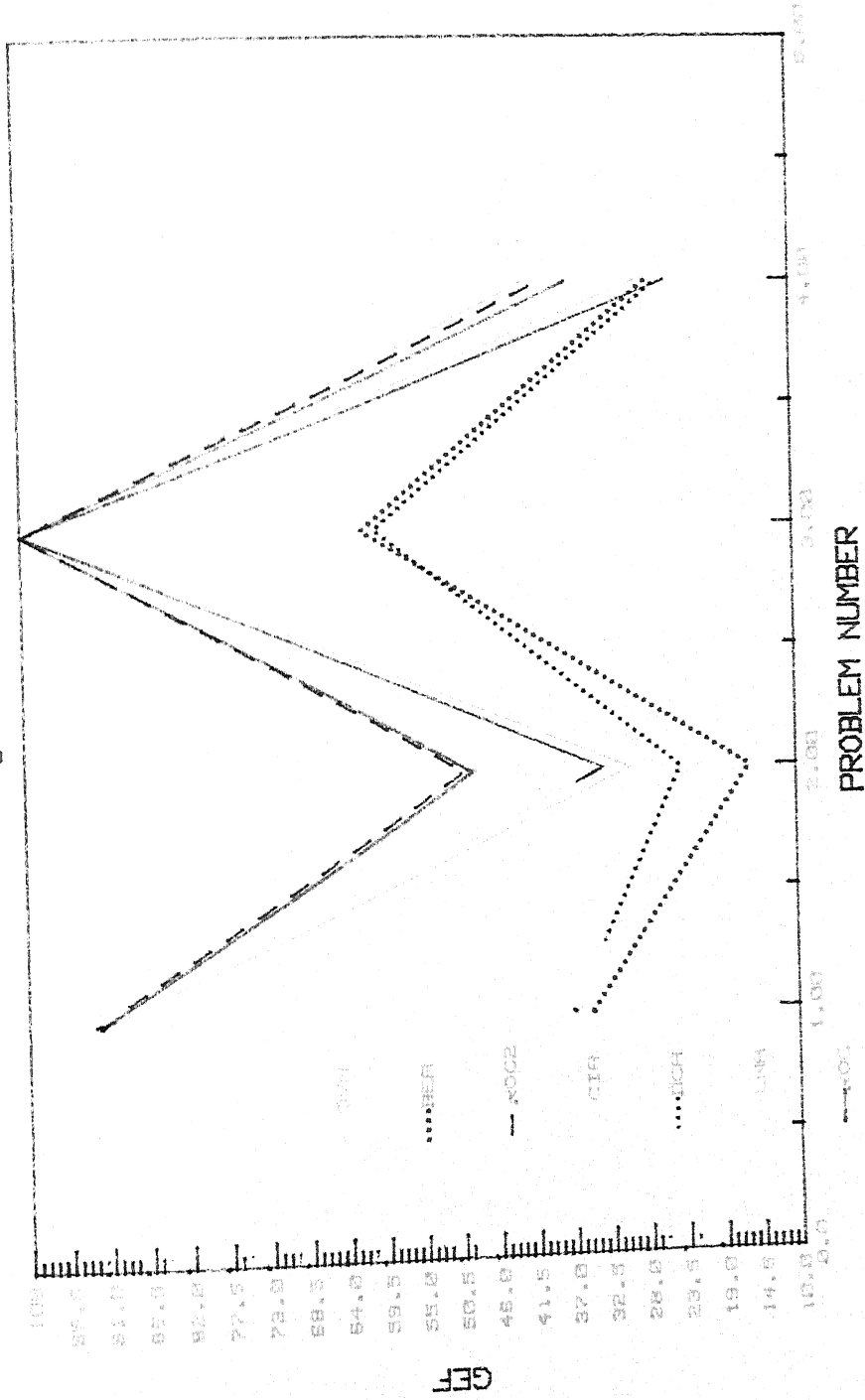
COMPARISON OF GROUPING EFFICIACY

Figure Number 5.35

GEF

PROBLEM NUMBER

Graphstar

COMPARISON OF GROUPING EFFICICACY

Figure Number 5.35

PROBLEM NUMBER

GEF

Graphstar
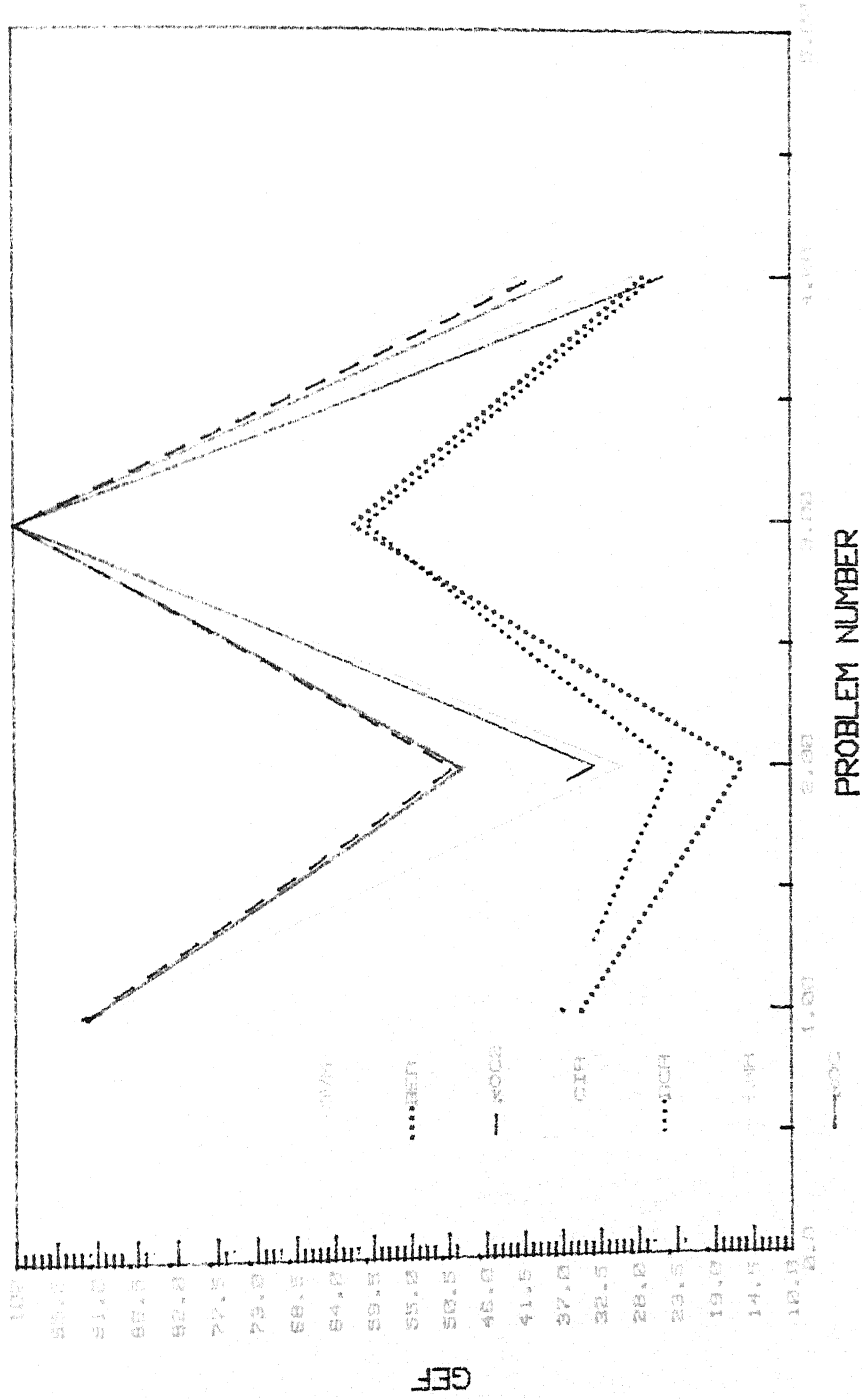
4.2 Observations

The following observations were made from the results

(1) Rank order clustering [39], extended rank order clustering [40] and occupancy value method [38] gave nearly equal grouping efficiencies.

(2) Cluster identification method [41] was successful in giving good grouping efficiency for some example, comparable with kings [39] and Occupancy value method, but it failed altogether for some examples. Cluster identification algorithm failed to identify clusters when machine part matrix was dense, and had more number of bottleneck machines and exceptional parts.

(3)The grouping efficiency of linear weighting algorithm [29] followed that of Rank order clustering and Occupancy value method, which was followed by Direct clustering algorithm [13] and then by bond energy algorithm [48].

(4) When a large matrix of 16 * 43 machines and parts was fed to the program, Rank order clustering [39] failed to group, because of integer overflow, whereas other methods gave answers. Thus Rank order clustering is not useful for larger matrices.

(5) For some examples Rank order clustering converged to solution which were not optimal but such occurrence were very few as compared to Linear weighting algorithm, Direct clustering algorithm and Bond energy algorithm.

(6) In Linear weighting algorithm the ranking values were sorted quickly and requirement of a very large integer representation does not arise.

(7) Linear weighting algorithm produced confusing patterns of the intermediate results together with the difficulty in predicting the behavior of the procedure.

(8) Linear weighting algorithm and Direct clustering algorithm were not able to produce exact diagonal matrices in most of the cases. The examination of resulting matrices may sometimes suggest one or two column or row interchanges to achieve a desired result.

(9) Bond energy algorithm gave solution matrix with a chequerboard structure with overlapping blocks.

(10) Bond energy algorithm gave more number of bottleneck machines than any other method.

(11) Direct clustering algorithm gave maximum number of exceptional parts followed by Bond energy algorithm.

(12) Rank order clustering, Extended Rank order clustering and Occupancy value method gave highest measure of effectiveness, it was followed by Cluster identification algorithm. Linear weighting algorithm, Direct clustering algorithm and Bond energy

algorithm gave relatively poorer measure of effectiveness.

(13) Cluster identification algorithm cannot deal with problems containing bottleneck parts.

(14) When we compare the CPU time Direct clustering algorithm, Bond energy algorithm, Linear weighting algorithm and Cluster identification algorithm takes less time as they involve less or no computations whereas Rank order clustering, Extended rank order clustering and Occupancy value method takes longer time as computations are involved.

4.3 Explanation

Rank order clustering [39] uses exponential weights and binary ranking. The reason why Rank order clustering gives good solution is that it assigns positional weights and that weight is of substantial order. There are less chances of ties in ranking. Mostly the tie will be only in case of exactly similar rows or columns. Thus the possibility of alternate optimum solution possibility is less. The positional weight gives justice to all the elements and binary ranking becomes discriminatory.

In some cases Rank order clustering does not give optimal results. This may be because of ties in columns and rows which allows alternate and better solution if rows and columns are rearranged. Rank order clustering method may not produce consistent results, when rows or columns in the initial matrix are interchanged. Thus Rank order clustering depends on initial

algorithm gave relatively poorer measure of effectiveness.

(13) Cluster identification algorithm cannot deal with problems containing bottleneck parts.

(14) When we compare the CPU time Direct clustering algorithm, Bond energy algorithm, Linear weighting algorithm and Cluster identification algorithm takes less time as they involve less or no computations whereas Rank order clustering, Extended rank order clustering and Occupancy value method takes longer time as computations are involved.

4.3 Explanation

Rank order clustering [39] uses exponential weights and binary ranking. The reason why Rank order clustering gives good solution is that it assigns positional weights and that weight is of substantial order. There are less chances of ties in ranking. Mostly the tie will be only in case of exactly similar rows or columns. Thus the possibility of alternate optimum solution possibility is less. The positional weight gives justice to all the elements and binary ranking becomes discriminatory.

In some cases Rank order clustering does not give optimal results. This may be because of ties in columns and rows which allows alternate and better solution if rows and columns are rearranged. Rank order clustering method may not produce consistent results, when rows or columns in the initial matrix are interchanged. Thus Rank order clustering depends on initial

disposition of the matrix. This occurs because, by changing rows and columns we are changing the positional weights which leads to different answers.

The occurrence of exceptional elements is expected but Rank order clustering solution is disrupted, due to the method adopted for ranking. It relies on pair wise comparisons of cell entries in the leftmost column (when ranking rows) and topmost row (when ranking columns). So if the positional occurrence of these elements is such that they influence the ranking, poor cluster formation will result. Bottleneck machines are used by a large number of components. Since these components can be expected to be dispersed over more than one cluster, such machine must appear in more than one row in the matrix. Otherwise the ranking procedure creates large, dispersed clusters with many machines and components contained in them. Rank order clustering algorithm will work more effectively after bottleneck machines and exceptional parts are identified and suppressed after visual analysis of the initial matrix solution. Such prior assumptions bias the solution, especially as the algorithm must indicate exceptions and bottleneck machines, not rely on their temporary suppression to be effective. The reason why sometimes Rank order clustering fails is because king [39] assumes that bottleneck machines can be freely duplicated if they are required in several clusters. He ignores the fact that some clusters would need to be merged to optimize the utilization of these machines.

The reason why Rank order clustering fails for larger

matrices is because of the procedure of reading the entries as binary words which presents some computational difficulties, since the largest integer representation in most computers is $(2^{48} - 1)$ or less, the maximum number of rows or columns that could be dealt with this way would be 47. Also the storage of the incidence matrix as a two dimensional array puts a severe limit on the problem that can be tackled. A moderate problem with 50 machines and 2000 components, together with the program would require core storage in excess of 120 K. Secondly because the sorting procedure has a complexity of cubic order, efficient implementation is not possible for very large problems. Extended Rank order clustering [40] was proposed to overcome this limitation. It uses the observation that GT problems are usually very sparse, with densities unlikely to be higher than 5 to 10 % . Extended Rank order clustering uses elaborate system of linked list structures and is very successful. Thus extended Rank order clustering gives answers when Rank order clustering fails. Further Rank order clustering and extended Rank order clustering uses array manipulations which reduces their efficiencies. However Rank order clustering gave good results comparable with Occupancy value method [38].

Occupancy value method gives good results because

(a) It creates compact clusters without any preliminary assumptions or visual identification of exceptional elements or bottleneck machines.

(b) It builds up clusters along the diagonal by using small selected sections of the larger original matrix. This allows the

analyst flexibility to re-iterate when ever dispersion of cell entries outside a cluster is observed, due to the occurrence of bottleneck machines. So, given any seed or starting machine, its components will indicate the other machines that will occur in the cluster, only a good choice of seed machine is required.

(c) It uses the machine component matrix only as a means for data representation and not for array manipulations like the Rank order clustering,Linear weighting algorithm, and Direct clustering algorithm.

(d) It uses the occupancy value to delay the entry of bottleneck machines and components with a large number of machines in their routes into the matrix. Otherwise this will create large clusters, useless for transforming into cells.

(e) It simplifies the identification of both exceptional elements and bottleneck machines by grouping correctly all the machines which would occur in only one cluster.

In CI (Kusiak [41]) the results obtained are good, but for many matrices with more density and more exceptional parts and bottleneck machines, the algorithm fails. This is because of the nature of algorithm which uses line drawing procedures. If matrix is dense all the rows and columns gets crossed resulting in one cell, the same as original matrix.

In Linear weighting algorithm (Graham [29]) the weighting of positions is linear. The weights increases linearly. This method can solve large matrices because requirement of a very large

obtained are inferior to Rank order clustering and Occupancy value method because it uses small positional weight for rows and columns. Hence ranking may involve ties. This leads to confusing patterns of the intermediate results together with the difficulty in predicting the behavior of the procedure. This method produces results inferior to Rank order clustering and Occupancy value method but better than Direct clustering algorithm and Bond energy algorithm.

In Direct clustering algorithm we do not use positional weighting procedure. We sum the entries of ones for each row and column and sort it according to their sum value. Direct clustering algorithm does not work properly because of ties in ranking which is very predominant. Adjustment of rows and columns may yield better results. It has all the difficulties of Rank order clustering except the storage.

The number of entries in the rows and columns which will increase in the matrices analyzed later, creating cluster dispersion similar to the influence of the bottleneck machines and exceptional elements. This method allows more flexibility in the size of problem. Furthermore, the sensitivity of the Rank order clustering algorithm to the initial matrix is eradicated because Direct clustering algorithm initiates the procedure by counting the number of positive cells instead of depending on intuition.

Bond energy algorithm [48] is applicable to problems of any

size because Bond energy algorithm has nothing to do with calculating the binary values. However the first step (select row or column) is determined by intuition. Many possible solutions can be generated, i.e the solution depends on the initial row or column selected for starting the process. Bond energy algorithm gives chequer board structure because the main aim of the algorithm is to improve measure of effectiveness and not to obtain a diagonal structure. The algorithms groups ones but they are not grouped in diagonal manner only the close proximity of the ones is enough to converge.

CHAPTER VI.

CONCLUSIONS

6.1 Conclusion

In this thesis role of group technology in booasting the productivity and its implications on the dynamic business environment is discussed. The concept of cell formation and division of system into subsystem has been discussed at appropriate places. A comprehensive literature survey has been presented and observations has been made on literature available. It has been stressed that although vast literature is available on group technology, comparison of different techniques on common scales has not been made. Many algorithms are explained in details giving the examples. These algorithms are both array type and heuristic algorithms. The algorithms are coded and results for cell formation are obtained therefrom. Four representative examples are choosen and algorithms are compared on basis of various measures like grouping efficiency, grouping efficiacy, measure of effectiveness, improvement in measure of effectiveness, time, number of bottleneck machines and exceptional parts. This measures of comparison are discussed in a seperate chapter. The results, tables and graphs are used as an aid for comparison. It has been found that rank order clustering, extended rank order clustering, occupancy value method and cluster identification method gives similar results which are better than direct clustering algorithm, linear weighting algorithm, and bond energy algorithm. The observations,

explanations and why different algorithms behave in a particular way, are given. Thesis ends with an observation that much can be done in this field by comparing more and more algorithms to find the best and suitable technique.

5.2 Scope for further research

Despite numerous economic benifits and operational advantages offered by the group technology, its real potential has not been fully explored.

Alternative approaches using recent techniques such as nueral nets, genetic algorithms, tabu search etc., in obtaining machine cells can be examined. Work can be done to integrate cell formation techniques with other functions of manufacturing organization such as design, process planning, production control, materials requirements planning, computer integrated manufacturing systems etc.

Various types of flexibilities can be incorporated in design of cellular manufacturing system. Use of goal programming for solving multiple objectives can be analysed. Probabilistic approaches, fuzzy theoretic approaches, polyhedral dynamic approaches, and expert system can be used in group technology. There is a need to exploit the power of simulation by simulating potential cellular manufacturing designs, before a decision is made on the selection of a particular design. Economic justification can be incorporated into group technology and more managerial issues can be analysed.

REFERENCES

[1]   Askin, R.G., and Chiu, K.S., 1990, A graph partitioning procedure for machine assignment and cell formation in group technology. International journal of production research, 28, 1555.

[2]   Askin, R.G., and Subramanyam, S., 1986, A cost based heuristic for group technology configuration, Working paper #86-006, University of Arizona.

[3]   Askin, R.G., Cresswell, S.H., Goldberg, J.B and Vakharia, A.J., 1991, A hamiltonian path approach to reordering the part machine matrix for cellular manufacturing, International journal of production research, 29, 1081.

[4]   Ballakur, A. and Steudel H.J., 1987, A within cell utilization based heuristic for cellular manufacturing systems, International journal of production research, 25, 639.

[5]   Banerjee, A., and Flynn B.B.,1987, A simulation study of some maintenance policies in a group technology shop, International journal of production research, 25, 1595.

[6]   Boctor,F.F., 1991, A linear formulation of the machine part cell formation problem, International journal of production research, 29, 343.

[7]   Boe, W.J., and Cheng, C.H., A close neighbour algorithm for designing cellular manufacturing systems, International journal of production research 29, 2097.

[8]   Burbridge, J.L., 1963, Production flow analysis, Production engineer 42, 742.

[9]   Burbidge, J.L.,1971 Production flow analysis, Production engineering, 50, 139-152.

[10]   Burbridge, J.L., 1973, Production flow analysis on the computer, Third Annual Conference of the Institution of Production Engineers.

[11]   Burbridge, J.L., 1977, A manual method of production flow analysis, Production Engineer 56, 34.

[12]   Carrie, A.S.,1973, Numerical taxonomy applied to group technology and plant layout, International journal of production research, 11, 399-416.

[13]   Chan H.M., and Milner, D.A.,1982, Direct clustering algorithm for group formation in cellular manufacture, Journel of manufacturing systems, 1, 65-74.

[14]   Chandrasekharan, M.P., and Rajagopalan, R., 1986a, An ideal seed non-hierarchical clustering for cellular manufacturing. International journal of production research 24, 451-464.

[15]   Chandrasekharan, M.P., and Rajagopalan, R., 1986b, MODROC: An extension of rank order clustering for group technology, International Journal of production Research, 24, 1221-1233.

[16]   Chandrasekaran, M.P., and Rajagopalan, R., 1987, ZODIAC- An algorithm for concurrent formation of part families and machine cells, International journal of production research, 25, 835-850.

[17]   Chandrasekharan, M.P., and Rajagopalan, R., 1989, GROUPABILITY : an analysis of the properties of binary data matrices for group technology, International Journal of production research 27, 1035.

[18]   Chandrasekharan, M.P and Suresh kumar, C., 1990, Grouping efficacy: a quantitative criterion for goodness of block diagonal forms of binary matrices in group technology, International journal of production research 28, 233.

[19]   Choobineh, F., 1988, A framework for the design of cellular manufacturing system, International journal of production research, 26, 1161.

[20]   Chu, C.H., and Hayya, J.C., 1991, A fuzzy clustering approach to manufacturing cell formation, International journal of production research 29, 1475.

[21]   Chu, C.H., and Tsai, M., 1990, A comparison of three array-based techniques for manufacturing cell formation, International journal of production research 28, 1417.

[22]   De Beer, C., and De Witte, J., 1978, Production flow synthesis, Ann. CIRP, 27, 389.

[23]   De Witte, J., 1979, The use of similarity coefficients in production flow analysis, Fifth International Conference on Production Research, 36-39.

[24]   De Witte 1980, The use of simalirity coefficients in production flow analysis, International journal of production research, 18, 503-514.

[25]   Dutta, S.P., Lashkari, R.S., Nadoli, G., and Ravi, T., 1986, Heuristic procedure for determining manufacturing families from design based grouping for flexible manufacturing systems. Computers and Industrial engineering 10, 193.

[26]   El Essawy, I.G.K., and Torrance, J., 1972, Component flow analysis -an effective approach to production systems design, Production engineering 51, 165-170.

[27]   Flynn, B.B., Jacobs, F.R.,1986, A simulation comparision of group technology with traditional job shop manufacturing , International journal of production research 24, 1171.

[28]   Flynn, B.B., 1987, The effects of setup time on output capacity in cellular manufacturing, International journal of

production research 25, 1761.

[29]   Graham, I., Galloway, P., and Scollar, I., 1976, Model studies in computer seriation, Journal of Archaeological Science 3.

[30]   Gunasingh, K.R., and Lashkari, R.S., 1989a, Machine grouping problem in cellular manufacturing systems-an integer programming approach, International Journal of Production Research, 27, 1465.

[31]   Gunasingh, K.R., and Lashkari, R.S., 1989b, The cell formation problem in cellular mfg systems A sequential modelling approach, Computers and Industrial engineering 16, 469.

[32]   Gunasingh, K.R., and Lashkari, R.S., 1991, Simultaneous grouping of parts and machines in cellular manufacturing systems - An integer programming approach, Computers and Industrial engineering. 20, 111.

[33]   Gupta, T., and Seifoddini, H., 1990, Production data based similarity coefficient for machine component grouping decisions in the design of cellular manufacturing system, International journal of production research, 28, 1247.

[34]   Gupta, T., and Seifoddini, H., 1990, Clustering algorithms for the design of a cellular manufacturing system - An analysis for their performance, Computers and Industrial engineering 19, 432.

[35]   Harhalakis, G., Nagi, R., and Proth, J.M., 1990, An efficient heuristic in manufacturing cell formation for group technology applications, International journal of production research 28, 185.

[36]   Hyer, N.L., and Wemmerlov, U., 1989, Group technology in the U.S. manufacturin industry: a survey of current practices, International journal of production research 27, 1287-1304.

[37]   Iri, M., 1968, On the synthesis of loop and cutset matrices and the related problems, SAAG Memoirs, 4, A-XIII, 376.

[38]   Khator, S.K., and Irani, S.A., 1987, Cell formation in group technology a new approach, Computers and Industrial engineering 12, 131.

[39]   King, J.R., 1980, Machine component grouping in production flow analysis: an approach using a rank order clustering algorithm, International journal of production research 18, 213.

[40]   King, J.R., and Nakornchai, V., 1982, Machine component group formation in group technology review and extension, International journal of production research 20, 117.

[41]   .. Kusiak, A., 1987a, The generalized group technology concept, International journal of production research, 25, 561.

[42]   Kusiak, A., 1988, EXGT-S: A knowledge based system for

group technology, International Journal of Production Research, 26, 887.

[43]   Kusiak, A., 1991, Branching algorithms for solving the group technology problem, Journal of manufacturing system, 10, 332.

[44]   Logendran, R., 1990, A workload based model for minimizing total intercell and intracell moves in cellular manufacturing, International journal of production research 28, 913.

[45]   Logendran, R., 1991, Impact of sequence of operations and layout of cells in cellular manufacturing. International journal of production research 29, 375.

[46]   Logendran, R., and West, T.M., A machine-part based grouping algorithm in cellular manufacturing, Computers and industrial engineering 19, 57.

[47]   McAuley, J., 1972, Machine grouping for efficient production, Production engineer, 51, 53-57.

[48]   McCormick, W.T., Schweitzer, P.J., and White, T.E., 1972, Problem decomposition and data recognition by a clustering technique, Operations Research 20, 993.

[49]   Nagi, R., Harhalakis, G., and Proth, J.M., 1990, Multiple routeings and capacity considerations in group technology applications, International journal of production research 28, 2243.

[50]   Offodile, O.F., 1991, Application of similarity coefficient method to parts coding and classification analysis in group technology, Journal of manufacturing systems 10, 442.

[51]   Purcheck, G.F.K., 1974, Combinatorial Grouping-a lattice theoritic method for the design of manufacturing systems, J. Cybernetics, 4, 27.

[52]   Purcheck, G.F.K., 1975a, A mathematical classification as a basis for the design of group technology production cells, Production Engineer 54, 35.

[53]   Purcheck, G.F.K., 1975b, A linear programming method for the combinatoric grouping of an incomplete power set, J. Cybernetics, 5, 51.

[54]   Rajagopalan, R., and Batra, J.L., 1975, Design of cellular production systems- a graph theoretic approach, International journal of production research 13, 567.

[55]   Robinson, D., and Duckstein, L., 1986, Polyhedral dynamics as a tool for machine-part group formation, International Journal of production Research, 24, 1255.

[56]   Sassani, F., 1990, A simulation study on performance improvement of group technology cells, International Journal of Production Research, 28, 293.

[57]   Seifoddini, H., 1990. A probibilistic model for machine cell formation, Journel of manufacturing Systems, 9, 69.

[58]   Shtub, A., 1989, Modelling group technology cell formation as a generalised assignment problem, International journal of production research, 27, 775.

[59]   Sinha, R.K., and Hollier, R.H., 1984, A review of production control problems in cellular manufacture. International journal of production research 22, 773-789.

[60]   Srinivasan, G., Narendran, T.T., and Mahadevan, B., 1990, An assignment model for the part families problem in group technology, International journal of production research 28, 145.

[61]   Steudel, H.J., and Ballakur, A.,1987, A dynamic programming based heuristic for machine grouping in manufacturing cell formation, Computers and Industrial engineering 12, 215.

[62]   Steudel, H.J., 1986, SIMSHOP: A job shop/cellular manufacturing simulator, Journal of manufacturing systems 5, 181.

[63]   Sule, D.R., 1991, Machine capacity planning in group technology. International journal of production research 29, 1909.

[64]   Tam, K.Y., 1990, An operation sequence based similarity coefficient for part families formulations, Journal of manufacturing systems 9, 55.

[65]   Vakharia, A.J.,and wemmerlov, U.,1990, Designing a cellular manufacturing system: A materials flow approach based on operation sequences International journal of production research 22, 84.

[66]   Vanelli, A., and Ravikumar, K., 1986, A method for finding minimimal bottleneck cells for grouping part-machine families, International journal of production research 24, 387.

[67]   Vohra, T., Chen, D.S., Chang, J.C., and Chen H.C.,. 1990, A network approach to cell formation in cellular manufacturing, International journal of production research 28, 2075.

[68]   Waghodekar, P.H., and Sahu, S., 1984, Machine component cell formation in group technology: MACE, International journal of production research 22, 937.

[69]   Wemmerlov, U., and Hyer, N.L.,1987, Research issues in cellular manufacturing, International journal of production research 25, 413-431.

[70]   Wemmerlov, U., and Hyer, N.L., 1989, Cellular manufacturing in the U.S. industry: a survey of users, International journal of production research 27, 1511-1530.

[71]   Wei, J.C., and Gaither, N., 1990, An optimal model for cell formation decisions, Decision sciences, 21, 416.

[72]   Wei, J.C., and Kern, G.M., 1989, Commonity analysis : A linear clustering algorithm for group technology, International Journal of production research 27, 2053.

[73]   Xu, H., and Wang, H.P., 1989, Part family for GT applications based on fuzzy mathematics, International journal of production research 27, 1637-1651.

APPENDIX A

A.1

This section gives the illustration for the Kings Rank order clustering [39] method. The original matrix given is as follows (Figure 1) on which the computations are done to obtain the final matrix. Further computions are done on the following figures.

Binary weight	6 2	5 2	4 2	3 2	2 2	1 2	0 2	decimal weight	binary ranking
0	1	2	3	4	5	6	7		
1	0	1	0	1	1	1	0	46	4
2	1	0	1	0	0	0	0	80	2
3	1	0	1	0	0	0	1	81	1
4	0	1	0	1	0	1	0	42	5
5	1	0	0	0	0	0	1	65	3

Figure 1 : Initial Matrix and ROC Calculations

0	1	2	3	4	5	6	7
3	1	0	1	0	0	0	1
2	1	0	1	0	0	0	0
5	1	0	1	0	0	0	1
1	0	1	0	1	1	1	0
4	0	1	0	1	0	1	0

Decimal weight

2 8	0 2	2 8	0 3	0 2	0 3	2 0

Figure 2 : Calculation for ROC Method

```
    0  1  3  7  2  4  6  5
   ┌─────────────────────────┐
 3 │ 1  1  1  0  0  0  0 │
 2 │ 1  1  0  0  0  0  0 │
 5 │ 1  0  1  0  0  0  0 │
 1 │ 0  0  0  1  1  1  1 │
 4 │ 0  0  0  1  1  1  0 │
   └─────────────────────────┘
```

Figure 3 : Final Matrix by ROC Method

Figure 3 is the final matrix obtained by the method.

A.2

This section illustrates the grahams linear weighting algorithm. the same illustration is used for solving by this method.

```
weight            7  6  5  4  3  2  1    linear weight

         0  1  2  3  4  5  6  7
        ┌─────────────────────┐
      1 │ 0  1  0  1  0  1  0 │      15
      2 │ 1  0  1  0  0  0  0 │      12
      3 │ 1  0  1  0  0  0  1 │      13
      4 │ 0  1  0  1  0  1  0 │      12
      5 │ 1  0  0  0  0  0  1 │       8
        └─────────────────────┘
```

Figure 4 : Calculation of LWA Method

0	1	2	3	4	5	6	7	weight
1	0	1	0	1	1	1	0	5
3	1	0	1	0	0	0	1	4
2	1	0	1	0	0	0	0	3
4	0	1	0	1	0	1	0	2
5	1	0	0	0	0	0	1	1
linear weight	8	7	7	7	5	7	5	

Figure 5 : Calculation of LWA Method

0	1	2	3	4	6	5	7	linear weight
1	0	1	0	1	1	1	0	15
3	1	0	1	0	0	0	1	13
2	1	0	1	0	0	0	0	12
4	0	1	0	1	1	0	0	13
5	1	0	0	0	0	0	1	8

Figure 6 : Calculation of LWA Method

0	1	2	3	4	6	5	7
1	0	1	0	1	1	1	0
3	1	0	1	0	0	0	1
4	0	1	0	1	1	0	0
2	1	0	1	0	0	0	0
5	1	0	0	0	0	0	1
linear weight	7	8	6	8	8	5	5

Figure 7 : Calculation of LWA Method

```
   0  2  4  6  1  3  5  7      linear weight

   1 | 1  1  1  0  0  1  0 |        20

   3 | 0  0  0  1  1  0  1 |         8

   4 | 1  1  1  0  0  0  0 |        18

   2 | 0  0  0  1  1  0  0 |         7

   5 | 0  0  0  1  0  0  1 |         5
```

Figure 8 : Calculation of LWA Method

```
   0  2  4  6  1  3  5  7

   1 | 1  1  1  0  0  1  0 |

   4 | 1  1  1  0  0  0  0 |

   3 | 0  0  0  1  1  0  1 |

   2 | 0  0  0  1  1  0  0 |

   5 | 0  0  0  1  0  0  1 |
```

Figure 9 : Block Diagonal Structure by LWA Method

A.3

This section illustrates chan and Milners, Direct Clustering Algorithm [13] with the same example.

```
   0  1  2  3  4  5  6  7        sum

   1 | 0  1  0  1  1  1  1 |        4

   2 | 1  0  1  0  0  0  0 |        2

   3 | 1  0  1  0  0  0  1 |        3

   4 | 0  1  0  1  0  1  0 |        3

   5 | 1  0  0  0  0  0  1 |        2
```

Figure 10 : Calculation by DCA Method

	0	1	2	3	4	5	6	7
2		1	0	1	0	0	0	0
5		1	0	0	0	0	0	1
3		1	0	1	0	0	0	1
4		0	1	0	1	0	1	0
1		0	1	0	1	1	1	0
sum		3	2	2	2	1	2	2

Figure 11 : Calculation by DCA Method

	0	1	7	3	4	6	2	5
2		1	0	1	0	0	0	0
5		1	1	0	0	0	0	0
3		1	1	1	0	0	0	0
4		0	0	0	1	1	1	0
1		0	0	0	1	1	1	1

Figure 12 : Block Diagonal Structure by DCA Method

A.4

This section illustratres the Cluster identification algorithm of Kusiak [41]. Selecting the first row as the selected row in the figure 1 we apply line drawing procedure to get initial machine set as machines {1, 4} and the part set as the parts {2,4,5,6}. Further choosing the machine 2 and applying the line drawing procedure we get machine set {2,3,5} and the part set as {1,3,5}. the final matrix obtained is

```
   0   2   4   5   6   1   3   7
  ┌─────────────────────────────┐
1 │ 1   1   1   1   0   0   0   │
  │                             │
4 │ 1   1   0   1   0   0   0   │
  │                             │
2 │ 0   0   0   0   1   1   0   │
  │                             │
3 │ 0   0   0   0   1   1   1   │
  │                             │
5 │ 0   0   0   0   1   0   1   │
  └─────────────────────────────┘
```

Figure 13 : Block Diagonal Structure by CIA Method

A.5

This section illustrates the Kings ROC2 algorithm [40]. The algorithm is applied on figure 1.

Row list

For column no.

```
┌──────────────────────┐
│  7   1   2   3   4   5 │
│                       │
│  6   3   5   1   2   4 │
│                       │
│  5   1   4   3   5   2 │
│                       │
│  4   1   4   3   5   2 │
│                       │
│  3   1   4   3   5   2 │
│                       │
│  2   3   2   1   4   5 │
│                       │
│  1   1   4   3   2   5 │
└──────────────────────┘
```

New row order   3   2   5   1   4

Figure 14 : Calculation of ROC2 Method

column list

```
For row no.    5  1  2  3  4  5  6  7

               4  2  4  6  1  3  5  7

               3  2  4  6  5  1  3  7

               2  1  7  2  4  6  5  3

               1  1  3  7  2  4  6  5
```

New column order   1 3 7 2 4 6 5

Figure 15 : Calculation by ROC2 Method

The final matrix obtained is as follows

```
0   1  3  7  2  4  6  5

3   1  1  1  0  0  0  0

2   1  1  0  0  0  0  0

5   1  0  1  0  0  0  0

1   0  0  0  1  1  1  1

4   0  0  0  1  1  1  0
```

Figure 16 : Block Diagonal Structure by ROC2 Method

A.6

This section illustrates the Bond energy method of McCormick[48]. The computations are done on figure 1.

Step 1 : Set j = 1. Select column 2.

Step 2 : Place each of the remaining columns in each of the j +1 positions. The contributions to the ME value of column 2 is computed next.

POSITION	j = 1	j + 1	ME value
	2	1	0
	2	3	0
COLUMN NUMBER	2	4	2
	2	5	1
	2	6	2
	2	7	0

Figure 17 : Calculation by BEA Method

Column 4 is placed in the j + 1 th position.

Repeating the same procedure for the remaining columns leads to the following column order {2,4,6,6,1,3,7}.

Step 3 : Repeating the preceding steps for rows results on the row order {1,3,2,4,5}. The final matrix obtained is

```
  0  2  4  6  5  1  3  7

  1 |1  1  1  1  0  0  0|

  3 |0  0  0  0  1  1  1|

  2 |0  0  0  0  1  1  0|

  4 |1  1  1  0  0  0  0|

  5 |0  0  0  0  1  0  1|
```

Figure 18 : Block Diagonal Structure by BEA Method

A.7

This section illustrates the method of solving a problem with the help of occupancy value method. The original matrix is given in figure 1.

ITERATION 1

Step 1 : Create an identical initial matrix [I] for use in all subsequent steps.

Step 2 : For each component j in [I] scan its column to find the total number of machines it uses M_j.

J	1	2	3	4	5	6	7
M_j	3	2	2	2	1	2	2

Step 3 : Scan M_j values in component list for selecting components with the minimum number of machines. Only one is found that is component 5.

Step 5 : Component 5 nad machine 1 enters the new matrix.

Step 6 : The initial matrix is now updated. Set all a_{ij} values in the rows correspomding to the entering machines zero and compute M_j values.

0	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
2	1	0	1	0	0	0	0
3	1	0	1	0	0	0	1
4	0	1	0	1	0	1	0
5	1	0	0	0	0	0	1
M_j	3	1	2	1	0	1	2

Figure 19 : Calculation by OVM Method

Iteration 2

Step 3 : Minimum M_j value is 1.

Step 4 : Since all three components require the same machine 4 occupancy value will be equal for all.

Step 5 : Enter any one of the component (say 2) and machine 4

into new matrix.

Step 6 : Update the matrix and find M_j.

0	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
2	1	0	1	0	0	0	0
3	1	0	1	0	0	0	1
4	0	0	0	0	0	0	0
5	1	0	0	0	0	0	1

M_j 3   0   2   0   0   0   2

Figure 20 : Calculation by OVM Method

ITERATION 3

Step 3 : Minimum M_j found is 2. There are ties so we go to step 4.

Step 4 : We find occupancy value for component 3

$I_1 = \{2,3\}$ $I_2 = \{5\}$ $J = \{1,3,7\}$

0	1	3	7
2	1	1	0
3	1	1	1
5	1	0	1

M_j 3   2   2

$$OV = \Sigma \, M_j / \, (m^* \, n)$$

$$= 7 / 9$$

$$= 0.777$$

Occupancy value for the component 7 will be same.

We enter component 3 and machines 2 and 3.

```
    0  1  2  3  4  5  6  7
    ┌─────────────────────┐
 1  │0  0  0  0  0  0  0  │
 2  │0  0  0  0  0  0  0  │
 3  │0  0  0  0  0  0  0  │
 4  │0  0  0  0  0  0  0  │
 5  │1  0  0  0  0  0  1  │
    └─────────────────────┘
```

Figure 21 : Calculation by OVM Method

Next we enter component 1 and 7 and machine 5, which yields full zero matrix. Final new matrix obtained is

```
    0  5  2  6  4  3  1  7
    ┌─────────────────────┐
 1  │1  1  1  1  0  0  0  │
 4  │0  1  1  1  0  0  0  │
 2  │0  0  0  0  1  1  0  │
 3  │0  0  0  0  1  1  1  │
 5  │0  0  0  0  0  1  1  │
    └─────────────────────┘
```

Figure 22 : Block Diagonal Structure by OVM Method

A.8

This section illustrates the method of calculating the grouping efficiency and grouping efficiacy.

Grouping efficiency :

$$\eta = q\eta_1 + (1 - q)\eta_2$$

$$\eta_1 = e_d / \sum M_r N_r = 14 / 17 = 0.8235$$

$$\eta_2 = 1 - e_0 / (mn - \sum M_r N_r) = 1$$

Taking q = 0.5 we get $\eta = 0.911$.

```
0  1  2  3  4  5  6  7

1  0  0  0  0  0  0  0

2  0  0  0  0  0  0  0

3  0  0  0  0  0  0  0

4  0  0  0  0  0  0  0

5  1  0  0  0  0  0  1
```

Figure 21 : Calculation by OVM Method

Next we enter component 1 and 7 and machine 5, which yields full zero matrix. Final new matrix obtained is

```
0  5  2  6  4  3  1  7

1  1  1  1  1  0  0  0

4  0  1  1  1  0  0  0

2  0  0  0  0  1  1  0

3  0  0  0  0  1  1  1

5  0  0  0  0  0  1  1
```

Figure 22 : Block Diagonal Structure by OVM Method

A.8

This section illustrates the method of calculating the grouping efficiency and grouping efficiacy.

Grouping efficiency :

$$\eta = q\eta_1 + (1 - q)\eta_2$$

$$\eta_1 = e_d / \sum M_r N_r = 14 / 17 = 0.8235$$

$$\eta_2 = 1 - e_0 / (mn - \sum M_r N_r) = 1$$

Taking $q = 0.5$ we get $\eta = 0.911$.

Grouping efficiacy :

$$GEF = (1 - \psi) / (1 + \phi)$$

ψ $=$ $\dfrac{\text{No. of exceptional elements}}{\text{Total no. of operations}}$ $= \dfrac{0}{14}$ $= 0.$

ϕ $=$ $\dfrac{\text{No. of voids in diagonal blocks}}{\text{Total no. of operations}}$ $= \dfrac{3}{14}$ $= 0.2143.$

GEF $=$ $1 / 1.2143$ $=$ $0.8235.$